



Polygon-to-polygon visibility : Computation, representation, applications

Frédéric Mora

► To cite this version:

Frédéric Mora. Polygon-to-polygon visibility : Computation, representation, applications. Autre [cs.OH]. Université de Poitiers, 2006. Français. NNT : . tel-00090127

HAL Id: tel-00090127

<https://theses.hal.science/tel-00090127>

Submitted on 29 Aug 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ACADÉMIE DE POITIERS
UNIVERSITÉ DE POITIERS
— SCIENCES FONDAMENTALES ET APPLIQUÉES —

THÈSE

pour l'obtention du Grade de
DOCTEUR DE L'UNIVERSITÉ DE POITIERS
(Faculté des Sciences Fondamentales et Appliquées)
(Diplôme National - Arrêté du 25 avril 2002)

École Doctorale : **Sciences Pour l'Ingénieur**
Secteur de Recherche : **Informatique**

Présentée par :

MORA FRÉDÉRIC

Visibilité polygone à polygone : calcul, représentation, applications

Soutenue le 10 juillet 2006 devant la Commission d'Examen composée de :

| | |
|---|----------------------|
| GHAZANFARPOUR Djamchid, Professeur des Universités, Université de Limoges | Rapporteur |
| PUECH Claude, Professeur des Universités, Université de Paris-Sud | Rapporteur |
| BOUATOUCH Kadi, Professeur des Universités, Université de Rennes I | Examineur |
| DECORET Xavier, Chargé de recherches, INRIA, INRIA Rhône-Alpes | Examineur |
| AVENEAU Lilian, Maître de Conférences, Université de Poitiers | Codirecteur de Thèse |
| MÉRIAUX Michel, Professeur des Université, Université de Poitiers | Directeur de Thèse |
| PAJUSCO Patrice, Responsable d'Unité R&D, France Telecom R&D Belfort | Invité |

Table des matières

| | |
|--|----------|
| Introduction | 1 |
| 1 Introduction à la visibilité | 5 |
| 1.1 Définitions et classification | 6 |
| 1.1.1 La visibilité en tant que preuve | 6 |
| 1.1.2 L'étude globale de la visibilité | 7 |
| 1.1.3 Positionnement | 11 |
| 1.2 Prérequis mathématiques | 13 |
| 1.2.1 Géométrie projective | 13 |
| 1.2.2 Paramétrisation de Plücker | 16 |
| 1.2.3 Polyèdres et polytopes | 17 |
| 1.3 Visibilité et espace de Plücker | 21 |
| 1.3.1 Visibilité à travers une séquence de polygones | 21 |
| 1.3.2 Prise en compte de l'occultation | 23 |
| 1.3.3 Événements visuels étendus | 24 |
| 1.4 Précédents travaux | 26 |
| 1.4.1 Arrangement d'hyperplans | 26 |
| 1.4.2 Le graphe d'aspect | 26 |
| 1.4.3 L'Asp | 27 |
| 1.4.4 Portails et antipénombre | 27 |
| 1.4.5 Le complexe de visibilité 2D | 28 |
| 1.4.6 Le complexe de visibilité 3D | 29 |
| 1.4.7 Le squelette de visibilité | 29 |
| 1.4.8 Arbre PSP | 30 |
| 1.4.9 Cellule de vue exacte | 31 |
| 1.4.10 Arbre d'occultation | 33 |
| 1.4.11 Preuve de visibilité | 35 |
| 1.5 Conclusion | 37 |

| | | |
|----------|---|------------|
| 2 | Visibilité polygone à polygone | 39 |
| 2.1 | Problématique | 40 |
| 2.1.1 | Objectifs | 40 |
| 2.1.2 | Effet de fragmentation | 40 |
| 2.2 | Approche proposée | 45 |
| 2.2.1 | Test d'intersection renforcé | 45 |
| 2.2.2 | Arbre VBSP | 47 |
| 2.2.3 | Valider et rentabiliser | 50 |
| 2.3 | Implantation et évaluation | 52 |
| 2.3.1 | Éléments d'implantation | 52 |
| 2.3.2 | Évaluation de la méthode | 53 |
| 2.3.3 | Discussion | 56 |
| 2.4 | Conclusion | 59 |
| 3 | Ombres douces en synthèse d'images | 61 |
| 3.1 | Calcul d'ombres douces | 62 |
| 3.1.1 | Subdivision de surfaces | 62 |
| 3.1.2 | Volumes d'ombre | 62 |
| 3.1.3 | Approches stochastiques | 63 |
| 3.1.4 | Faisceaux de rayons | 64 |
| 3.1.5 | Volumes d'ombre et de pénombre | 65 |
| 3.1.6 | Silhouette étendue et volume d'ombre douce | 68 |
| 3.1.7 | Maillage de discontinuités et événements visuels | 69 |
| 3.2 | Approche proposée | 70 |
| 3.2.1 | Vue d'ensemble | 70 |
| 3.2.2 | Visibilité point-polygone | 71 |
| 3.2.3 | Résultats | 74 |
| 3.3 | Discussion et perspectives | 81 |
| 3.3.1 | Extensions et optimisations | 81 |
| 3.3.2 | Limitations actuelles | 81 |
| 3.3.3 | Perspectives | 82 |
| 4 | Prédiction sans perte de la propagation des ondes électromagnétiques | 85 |
| 4.1 | Préambule | 87 |
| 4.1.1 | Les enjeux | 87 |
| 4.1.2 | Caractéristiques du canal de propagation | 87 |
| 4.1.3 | Les modèles basés rayons | 88 |
| 4.2 | Approche proposée | 92 |
| 4.2.1 | Requête de visibilité unifiée | 92 |
| 4.2.2 | Graphe de visibilité : calcul hiérarchique mixte | 99 |
| 4.2.3 | Résultats | 104 |
| 4.3 | Conclusions | 108 |
| 4.3.1 | Discussion | 108 |
| 4.3.2 | Perspectives | 109 |
| | Conclusion | 111 |
| | Bibliographie | 115 |

Table des figures

Table des figures

| | | |
|------|---|----|
| 1.1 | Les bloqueurs de deux objets | 7 |
| 1.2 | Droites du plan et de l'espace et dimension associée | 7 |
| 1.3 | Illustration de l'étude globale de la visibilité | 8 |
| 1.4 | Droites poignardantes extrêmes et configurations géométriques associées | 9 |
| 1.5 | Événement visuel VE | 10 |
| 1.6 | Événement visuel EEE | 10 |
| 1.7 | Le plan projectif : illustration | 13 |
| 1.8 | Droite du plan projectif | 14 |
| 1.9 | Intersection de droites dans le plan projectif | 15 |
| 1.10 | Interprétation géométrique des coordonnées de Plücker | 16 |
| 1.11 | Orientation relative de deux droites dans l'espace de Plücker | 17 |
| 1.12 | Position d'un point et d'un hyperplan dans l'espace de Plücker : interprétation | 18 |
| 1.13 | H-représentation et V-représentation d'un polytope | 19 |
| 1.14 | Graphe d'incidence d'un polytope | 20 |
| 1.15 | Complexe de polytopes | 20 |
| 1.16 | Caractérisation des droites poignardant un triangle | 21 |
| 1.17 | Caractérisation des droites poignardant une séquence de polygones | 22 |
| 1.18 | Caractérisation de l'occultation dans l'espace de Plücker | 23 |
| 1.19 | Droites critiques des bords d'un complexe de polytopes | 24 |
| 1.20 | Visibilité à travers une séquence de polygones convexes | 28 |
| 1.21 | Illustration 2D de l'antipénombre | 28 |
| 1.22 | Illustration de la structure de squelette de visibilité | 30 |
| 1.23 | Événements visuels dégénérés | 30 |
| 1.24 | Polytope caractérisant la visibilité entre deux polygones : construction selon Nirenstein (1) | 32 |
| 1.25 | Polytope caractérisant la visibilité entre deux polygones : construction selon Nirenstein (2) | 32 |
| 1.26 | Illustration 2D de la structure d'arbre d'occultation | 34 |
| 1.27 | Polytope caractérisant la visibilité entre deux polygones : généralisation selon Haumont | 35 |
| 2.1 | Soustraction d'un ensemble de droites occultées | 41 |
| 2.2 | Opérations inutiles : premier exemple | 42 |

| | | |
|------|--|-----|
| 2.3 | Opérations inutiles : deuxième exemple | 42 |
| 2.4 | Rejet de la V-représentation d'un polytope par la H-représentation d'un polyèdre | 43 |
| 2.5 | Un cas de fragmentation de la visibilité | 43 |
| 2.6 | Esl-représentation d'un bloqueur | 45 |
| 2.7 | Test de rejet de l'esl-représentation d'un bloqueur par un polytope | 46 |
| 2.8 | Partition d'un ensemble de visibilité | 48 |
| 2.9 | Rentabiliser les opérations inutiles | 51 |
| 2.10 | Représentation BSP de la visibilité de deux polygones | 51 |
| 2.11 | Exemple de scènes pour la première série de tests | 54 |
| 2.12 | Résultats de la première série de tests | 55 |
| 2.13 | Aperçu de la scène pour la seconde série de tests | 56 |
| 2.14 | Résultats de la seconde série de tests | 57 |
| 3.1 | Volume d'ombre dure | 63 |
| 3.2 | Ombre douce : approche stochastique | 63 |
| 3.3 | Volume d'ombre et de pénombre | 65 |
| 3.4 | Méthode de Chin et Feiner par subdivisions de surfaces | 66 |
| 3.5 | Illustration de la méthode Laine | 67 |
| 3.6 | Arête silhouette pour une source surfacique | 68 |
| 3.7 | Fusion de bloqueur et arbre VBSP | 70 |
| 3.8 | Visibilité depuis un point à partir de la visibilité depuis un polygone | 71 |
| 3.9 | Extraction de la visibilité d'un polygone depuis un point | 72 |
| 3.10 | Calcul d'ombres douces : première scène de test | 75 |
| 3.11 | Calcul d'ombres douces : seconde scène de test | 76 |
| 3.12 | Calcul d'ombres douces : troisième scène de test | 77 |
| 3.13 | Calcul d'ombres douces : accroissement de la taille des sources | 79 |
| 4.1 | Principe de la méthode des source-images | 90 |
| 4.2 | Visibilité et bords d'un polytope | 93 |
| 4.3 | Graphe d'incidence inclu d'une k-face | 94 |
| 4.4 | Extraction de la visibilité des arêtes de deux polygones (1) | 96 |
| 4.5 | Extraction de la visibilité des arêtes de deux polygones (2) | 97 |
| 4.6 | Extraction de la visibilité des arêtes de deux polygones (3) | 98 |
| 4.7 | Nature d'un graphe de visibilité | 99 |
| 4.8 | Résultats de calcul de zones de couverture | 106 |

Introduction

La visibilité est une question incontournable dans plusieurs domaines de recherche. Ainsi, en synthèse d'images, elle est sous-jacente à de nombreuses problématiques. L'information de visibilité requise peut être ponctuelle ou surfacique. Dans le premier cas, il s'agira par exemple de calculer la vue d'un observateur virtuel, ou encore les ombres dures engendrées par une source lumineuse. Dans le second cas, se posent des problèmes plus complexes où l'information de visibilité relative aux polygones d'un environnement devient nécessaire. Par exemple, c'est le cas du calcul des ombres douces projetées par une source étendue. Cette dernière induit des zones d'ombre, de pénombre, et des zones totalement éclairées, qui correspondent respectivement à des régions invisibles, partiellement visibles, et totalement visibles depuis la source. Autrement dit, les ombres douces projetées sur un polygone dépendent de la visibilité de celui-ci avec la source étendue. Un autre exemple est celui des calculs d'illumination globale. Il s'agit alors de simuler les échanges d'énergie lumineuse au sein d'un environnement : tout polygone ayant reçu de la lumière se comporte à son tour comme une source émettrice. Les échanges d'énergie entre deux éléments de surfaces dépendent ainsi de leur visibilité mutuelle.

Le domaine des radiocommunications est également concerné par ces problèmes de visibilité : un enjeu majeur est la simulation de la propagation des ondes électromagnétiques, par exemple, afin de prédire la zone de couverture d'une antenne. Le principe est de simuler les interactions d'une onde électromagnétique avec un environnement donné. Dans ce cadre aussi, le flux échangé entre deux polygones dépend de leur visibilité mutuelle.

De manière générale, les mêmes problèmes de visibilité se posent pour tous les phénomènes physiques dont la propagation repose sur la notion de rayon. Les simulations sonores dans le domaine de l'acoustique en sont un autre exemple. Dans tous les cas, la visibilité mutuelle de polygones joue un rôle essentiel. La précision de cette information conditionne la précision des calculs propres à l'application, qu'il s'agisse de lumière ou d'énergie électromagnétique.

Cependant, calculer la visibilité est complexe. Les problématiques impliquant un point de vue sont aujourd'hui relativement bien maîtrisées. En revanche, la visibilité entre surfaces ou polygones demeure une question difficile. Comme nous l'avons vu, la précision de cette information est importante pour de nombreuses applications. Néanmoins, la grande majorité des approches pratiques ne procurent aujourd'hui

d’hui que des solutions partielles ou approximatives. C’est là le point de départ des travaux présentés dans ce mémoire.

Notre premier objectif est de proposer une solution adaptée au calcul, mais aussi à la représentation de la visibilité mutuelle de deux polygones. Cette étape franchie, nous explorons le potentiel offert par cette information afin de démontrer son intérêt dans la résolution de problèmes pratiques, à savoir le calcul des ombres douces en synthèse d’images, et la simulation de la propagation des ondes électromagnétiques en radiocommunications. À aucun moment nous ne ferons de concession sur la précision de l’information de visibilité calculée.

Contributions

Les contributions de cette thèse s’organisent en trois points. Le premier concerne le calcul d’une représentation de la visibilité mutuelle de deux polygones. Les deux suivants utilisent cette représentation pour imaginer de nouvelles solutions à deux problèmes bien connus : le calcul des ombres douces en synthèse d’images, et la propagation des ondes électromagnétiques en radiocommunication.

- Nous proposons un nouvel algorithme pour le calcul et la représentation dans l’espace de Plücker de la visibilité mutuelle de deux polygones. Ces travaux analysent et combinent les approches de Nirenstein [NBG02] et de Bittner [Bit02] tout en palliant leurs faiblesses. En particulier, notre algorithme améliore la pertinence des calculs réalisés, engendrant une diminution significative du temps de calcul et de la consommation mémoire. De plus, et dans la mesure où il n’existe que peu d’indications sur la complexité pratique de ce type d’approche, nous évaluons celle de notre algorithme et la confrontons aux meilleures bornes connues de sa complexité théorique.
- Pour la synthèse d’images, nous proposons une nouvelle solution au problème du calcul analytique des ombres douces. Celle-ci s’appuie sur l’information de visibilité de chaque polygone avec chaque source surfacique d’un environnement. Sur la base de cette information, nous présentons un algorithme d’extraction des parties visibles d’un polygone depuis un point. Intégré à un logiciel de rendu, cet algorithme permet un calcul exact de l’illumination directe en tous les points d’une scène. Les tests réalisés rendent compte de temps de calculs nettement inférieurs à ceux d’un logiciel stochastique commercial. Ces résultats s’accompagnent d’une étude pratique de notre algorithme d’extraction.
- En radiocommunications, nous présentons la première optimisation sans perte pour simuler la propagation des ondes électromagnétiques. Ces travaux se démarquent des autres méthodes qui ne peuvent produire que des simulations approchées, posant le problème fondamental de la fiabilité de leurs résultats. Notre approche exploite l’information de visibilité entre polygones pour extraire la visibilité de leurs arêtes. Elle établit hiérarchiquement toutes les relations de visibilité entre les polygones et les dièdres d’un environnement. Le logiciel résultant garantit l’exactitude des simulations et des temps de simulations raisonnables sur des environnements de taille significative.

Présentation du mémoire

Le premier chapitre constitue une introduction générale à la visibilité et aux principes fondamentaux sous-jacents à ce mémoire. Nous rappelons en premier lieu quelques notions élémentaires relatives à tout problème de visibilité. Puis, nous présentons les outils mathématiques et géométriques utilisés dans cette thèse. Ensuite, un état de l’art ciblé des travaux en rapport avec la représentation et le calcul de la visibi-

lité tridimensionnelle est présenté. Enfin, nous concluons en mettant en avant le potentiel sous-exploité des approches basées sur la paramétrisation de Plücker.

Le second chapitre est consacré au calcul et à la représentation dans l'espace de Plücker de la visibilité entre deux polygones. Ce chapitre correspond à la première des trois contributions présentées ci-dessus.

Le troisième chapitre présente notre approche du calcul des ombres douces en synthèse d'images. On y trouvera également un tour d'horizon des approches existantes, mettant en avant la difficulté de concilier qualité et rapidité du rendu.

Le quatrième chapitre regroupe nos travaux sur la propagation des ondes électromagnétiques en radio-communications. Nous présentons au préalable les principales problématiques propres à ce domaine de recherche, ainsi que les fondements théoriques qui permettent de valider le modèle de propagation que nous avons utilisé.

Enfin la conclusion résume l'ensemble des travaux présentés dans ce mémoire et dégage les principales perspectives de travail qu'il nous semble essentiel d'explorer à moyen terme.

Introduction à la visibilité

Sommaire

| | |
|--|-----------|
| 1.1 Définitions et classification | 6 |
| 1.1.1 La visibilité en tant que preuve | 6 |
| 1.1.2 L'étude globale de la visibilité | 7 |
| 1.1.3 Positionnement | 11 |
| 1.2 Prérequis mathématiques | 13 |
| 1.2.1 Géométrie projective | 13 |
| 1.2.2 Paramétrisation de Plücker | 16 |
| 1.2.3 Polyèdres et polytopes | 17 |
| 1.3 Visibilité et espace de Plücker | 21 |
| 1.3.1 Visibilité à travers une séquence de polygones | 21 |
| 1.3.2 Prise en compte de l'occultation | 23 |
| 1.3.3 Événements visuels étendus | 24 |
| 1.4 Précédents travaux | 26 |
| 1.4.1 Arrangement d'hyperplans | 26 |
| 1.4.2 Le graphe d'aspect | 26 |
| 1.4.3 L'Asp | 27 |
| 1.4.4 Portails et antipénombre | 27 |
| 1.4.5 Le complexe de visibilité 2D | 28 |
| 1.4.6 Le complexe de visibilité 3D | 29 |
| 1.4.7 Le squelette de visibilité | 29 |
| 1.4.8 Arbre PSP | 30 |
| 1.4.9 Cellule de vue exacte | 31 |
| 1.4.10 Arbre d'occultation | 33 |
| 1.4.11 Preuve de visibilité | 35 |
| 1.5 Conclusion | 37 |

1.1 Définitions et classification

Parce que c'est une notion naturelle et intuitive pour l'homme, le terme *visibilité* est souvent utilisé au sens large et sans nuance. Il englobe des travaux dont les objectifs méritent pourtant d'être distingués. Précisément, quelle est la preuve de la visibilité mutuelle de deux objets ? Qu'est-ce que la visibilité globale ? Ce sont ces différents points que nous précisons en premier lieu, afin de mieux positionner les travaux présentés dans ce mémoire.

1.1.1 La visibilité en tant que preuve

Dans cette section, sont définis différents termes et notions inhérents à notre contexte. Nous nous plaçons dans le cadre de scènes composées de polygones convexes. Les objets auxquels nous faisons référence sont des points, des segments, des polygones convexes, ou des polyèdres convexes.

D'une manière générale, un problème de visibilité est souvent ramené à la question : "Est-ce que c'est visible ?". Si l'on considère uniquement deux objets, l'ensemble de leur visibilité s'assimile à l'ensemble des droites qui les intersectent. Dans notre contexte, on parle alors souvent de *droites poignardantes* (*stabbing lines*). La définition 1 précise cette notion.

Définition 1 *Relativement à un ensemble d'objets $\{A_i, 1 \leq i \leq n\}$, une droite est dite poignardante si elle intersecte tous les objets de cet ensemble (en incluant les cas de tangence). On notera $stab(A_1, \dots, A_n)$ l'ensemble des droites poignardant tous les A_i .*

Plus généralement, la visibilité de deux objets se pose relativement à un ensemble d'obstacles que nous désignerons par *bloqueurs*. Dans notre contexte, un bloqueur satisfait à la définition suivante :

Définition 2 *Un objet O est un bloqueur pour deux objets A et B s'il possède une intersection non vide avec l'enveloppe convexe de A et de B .*

Trouver une visibilité entre deux objets revient à déterminer une droite qui les intersecte, sans intersecter un seul de leurs bloqueurs. On parle souvent de droite poignardant (*stabbing line*) ces objets.

Sur la base de ces définitions, la preuve de la visibilité mutuelle de deux objets s'interprète géométriquement comme suit :

Définition 3 *Etant donnés deux objets A et B ainsi que l'ensemble de leurs bloqueurs $\{O_i, 1 \leq i \leq n\}$, A et B sont visibles si, et seulement si, il existe au moins une droite $l \in stab(A, B)$ telle que $l \notin \bigcup_{i=1}^n stab(O_i)$.*

Toute droite satisfaisant à la définition 3 est la preuve que deux objets sont visibles. La figure 1.1 donne deux exemples. La majeure partie des travaux existants ont pour objet de prouver l'existence ou l'absence de visibilité. Cette information binaire est suffisante pour répondre à des problématiques, aujourd'hui classiques. Par exemple, quel est le premier objet rencontré par un rayon ? Quels sont les objets visibles depuis un point de vue ? Le classique lancer de rayons est à la base des solutions pratiques à ces problèmes, et à tous ceux qui s'y ramènent. Populaire et robuste, cette approche ne permet cependant pas de mieux comprendre les phénomènes visuels. Une simple information binaire de visibilité revient à explorer un monde obscur avec un pointeur laser. L'image souvent reprise est celle de l'étude de la visibilité en tant que fonction ou équation. La visibilité selon un rayon procure un ensemble de points satisfaisant à la fonction de visibilité. Mais elle ne procure pas d'information globale sur la fonction en elle-même. La seule preuve de la visibilité est insuffisante pour approfondir la compréhension globale du phénomène. Or, cette compréhension est nécessaire pour appréhender des questions plus complexes.

Par exemple : quelles sont toutes les vues possibles d'un objet ? Quelles sont les limites d'ombre et de

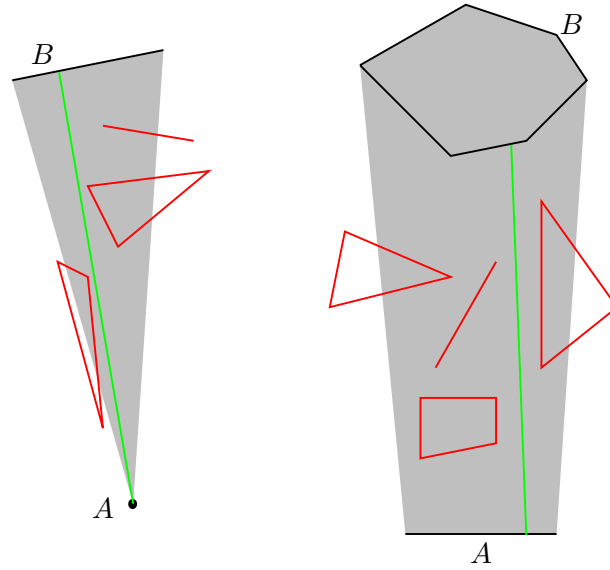


FIG. 1.1 – Les bloqueurs de deux objets A et B (points, segments, polygones) sont les éléments qui possèdent une intersection non vide avec l’enveloppe convexe de A et B . A et B sont visibles s’il existe une droite qui les intersecte mais qui ne possède pas d’intersection ou de relation de tangence avec un seul de leurs bloqueurs.

pénombre engendrées par une source surfacique ? Quelles sont toutes les visibilitées d’un environnement donné ? Ce type de problème conduit à une étude en compréhension de la visibilité. Les travaux en ce sens relèvent de la *visibilité globale*.

1.1.2 L’étude globale de la visibilité

La visibilité globale n’a pas pour objet premier la résolution d’un “problème de visibilité”. Il s’agit avant tout d’acquérir une meilleure connaissance du phénomène visuel. Elle ne se place pas pour autant hors du champ applicatif, bien au contraire. C’est un savoir complémentaire et nécessaire pour proposer des solutions toujours plus performantes à des problèmes concrets.

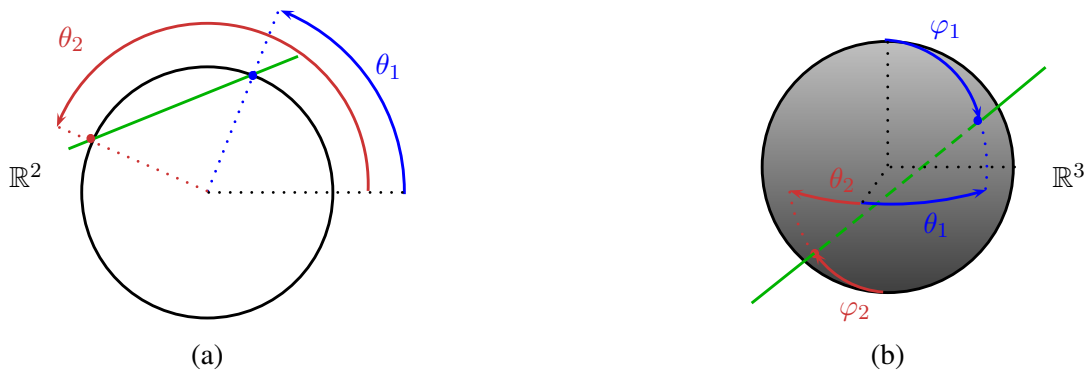


FIG. 1.2 – (a) : les droites du plan forment un espace de dimension 2. Toute droite est définie de façon unique par le couple (θ_1, θ_2) , relativement à ses points d’intersection avec le cercle circonscrit à la scène. (b) : les droites de \mathbb{R}^3 forment un espace de dimension 4. Toute droite est définie de façon unique par le quadruplet $(\theta_1, \varphi_1, \theta_2, \varphi_2)$, (θ_1, φ_1) et (θ_2, φ_2) désignant ses intersections avec la sphère englobante de la scène.

D'une manière générale, la visibilité globale se distingue par l'étude topologique de la visibilité, c'est-à-dire l'étude des lieux où se produit une discontinuité de la visibilité, ainsi que des relations de voisinage qui les lient. Intuitivement, il s'agit de regrouper les segments qui interceptent les mêmes objets à leurs extrémités. Ces regroupements constituent des cellules \mathcal{C}_{A_i, A_j} , chacune représentant une composante connexe de segments qui interceptent le même couple d'objets (A_i, A_j) . Ces segments peuvent aussi être considérés comme des rayons qui "voient" les mêmes objets dans les deux directions. L'enjeu est de les structurer vis-à-vis de leurs frontières respectives. C'est en ces frontières qu'ont lieu les changements dans la visibilité. Elles sont constituées de segments critiques, ou encore considérant leur droites supports, de *droites critiques* :

Définition 4 Une droite critique est une droite privée d'au moins un degré de liberté, car tangente à des objets de l'environnement considéré.

Une étude globale de la visibilité s'appuie généralement sur un espace dual (de droites ou de rayons) dans lequel les propriétés étudiées s'expriment le mieux. Dans \mathbb{R}^2 , une droite possède 2 degrés de liberté. En effet, deux paramètres suffisent à représenter une droite. Par exemple, il s'agira des deux points d'intersection de cette droite avec le cercle circonscrit à la scène comme illustré sur la figure 1.2.a. S'agissant alors d'une courbe paramétrée 1D, les droites du plan forment donc un espace 2D. Il en résulte que les droites critiques possèdent soit 1 degré de liberté, soit aucun.

Dans \mathbb{R}^3 , une droite possède 4 degrés de liberté. Le problème est plus ardu car l'espace des droites devient 4D. Pour conserver une image analogue au cas 2D, toute droite peut être définie de manière unique par ses deux points d'intersection avec la sphère englobante de la scène (voir figure 1.2.b). S'agissant alors d'une surface paramétrée 2D, les droites de \mathbb{R}^3 forment un espace 4D. Les ensembles de droites critiques sont donc de dimensions 0, 1, 2 ou 3. De la 2D à la 3D, le problème gagne donc 2 dimensions.

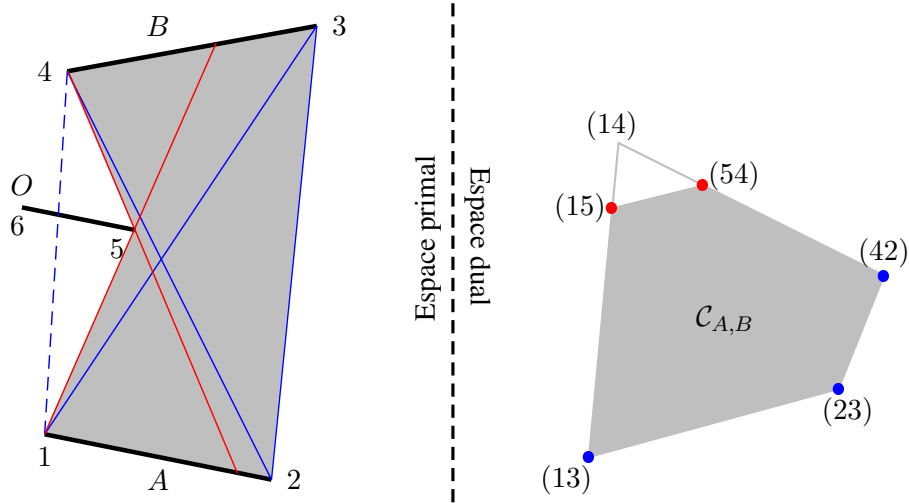


FIG. 1.3 – Illustration de l'étude globale de la visibilité. À gauche, on considère la visibilité de deux objets A et B . À droite, un espace dual est utilisé pour représenter l'ensemble des segments qui interceptent A et B à leurs extrémités. Ils constituent une cellule $\mathcal{C}_{A,B}$ dont les bords correspondent à des ensembles de droites critiques. Par exemple sur ce schéma, l'arête (13)-(15) de cellule $\mathcal{C}_{A,B}$ correspond aux visibilités depuis le sommet 1 du segment A dans l'espace primal. L'arête (15)-(14) correspond en revanche aux droites occultées par le segment O depuis le sommet 1.

Ainsi dans \mathbb{R}^2 les bords d'une cellule de dimension 2, *i.e.* représentant un ensemble 2D de segments (ou droites supports), sont formés de cellules de dimension 1 et 0 représentant des segments contraints

possédant 1 degré de liberté ou aucun. La figure 1.3 donne un exemple.

Dans \mathbb{R}^3 les bords d'une cellule de dimension 4, sont formés de cellules de dimensions 0 à 3. Par nature, la visibilité globale est un problème de dimension 2 pour des environnements 2D et de dimension 4 pour des environnements 3D. De part la dimension du problème, l'étude de la visibilité dans \mathbb{R}^3 n'est pas intuitive. Les ensembles critiques de dimensions 1 ou 0 ont fait l'objet d'une attention particulière. Ils sont certes plus simples à appréhender, mais surtout, ils jouent un rôle significatif dans différents problèmes de visibilité.

Les droites critiques de dimension 0 sont aussi appelées *droites poignardantes extrêmes* :

Définition 5 Une droite poignardante extrême ou *esl* (extremal stabbing line) est une droite qui ne possède aucun degré de liberté.

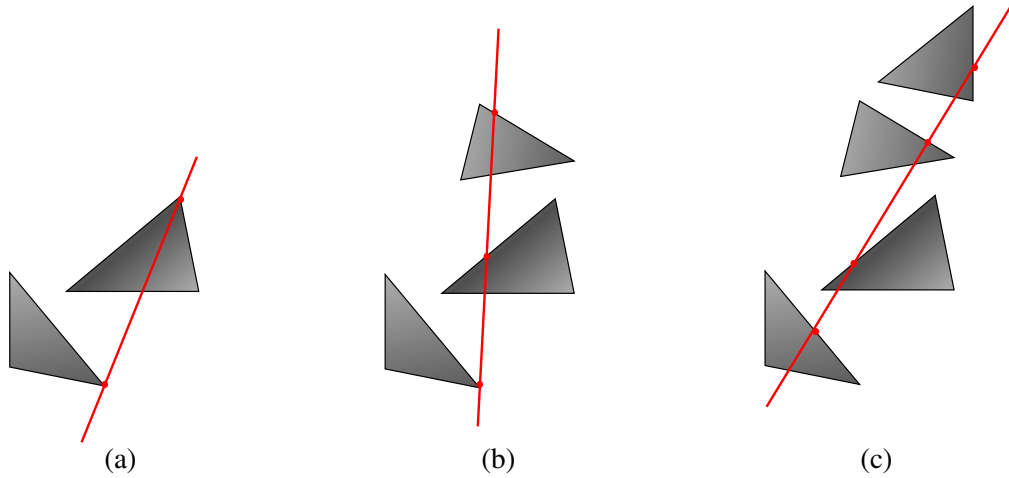


FIG. 1.4 – Dans \mathbb{R}^3 , 3 configurations peuvent engendrer une droite poignardante extrême(*esl*). (a) Un couple de sommets (VV). (b) Un sommet et deux arêtes (VEE). (c) Quatre arêtes (EEEE). Les cas (a) et (b) sont des cas particuliers du cas (c), un sommet étant incident à deux arêtes.

Quatre arêtes en position non dégénérée (*i.e.* non coplanaires deux à deux) peuvent engendrer une *esl* [TH99], *i.e.* une droite tangente à chacune d'entre elles. Plusieurs configurations géométriques engendrent une *esl*. Celles-ci sont présentées sur la figure 1.4. Les sommets (V - *Vertex*) et les arêtes (E - *Edge*) de la géométrie qui définissent une *esl* sont aussi appelés ses *générateurs*.

En relâchant l'une de 4 contraintes d'une *esl*, on obtient un ensemble de droites critiques de dimension 1. De tels ensembles décrivent dans l'espace une surface. Celle-ci est plane si ses générateurs sont dans une configuration dégénérée comme sur la figure 1.5, ou bien elle correspond à une surface réglée comme l'illustre la figure 1.6. Il est question de *surface critique*, de deux types possibles, VE dans le premier cas et EEE dans le second. Les surfaces critiques sont le lieu d'*événements visuels* :

Définition 6 Un événement visuel est un changement qualitatif de la visibilité, *i.e.* le lieu où un objet apparaît ou disparaît pour un observateur mobile qui traverse la surface associée à l'événement.

Les figures 1.5 et 1.6 donnent des exemples d'événements visuels survenant lorsque le regard d'un observateur franchit une surface critique VE ou EEE.

Les événements visuels ont été introduits en vision par ordinateur pour la reconnaissance d'objets. L'enjeu était de définir toutes les vues possibles d'un objet afin de pouvoir l'identifier quel que soit le point

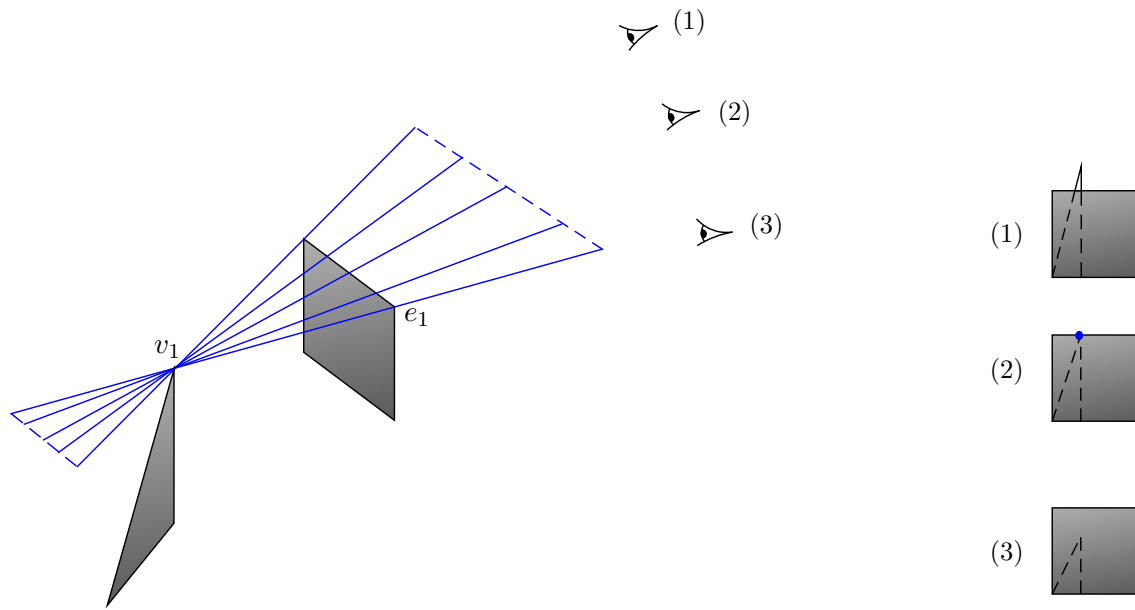


FIG. 1.5 – Un événement visuel VE (*Vertex-Edge*) est défini par le sommet et l'arête de deux polygones distincts (ici v_1e_1). La surface associée est plane. (1) Au dessus, le triangle est visible. (2) A la frontière, le sommet et l'arête se superposent visuellement. (3) En dessous le triangle disparaît.

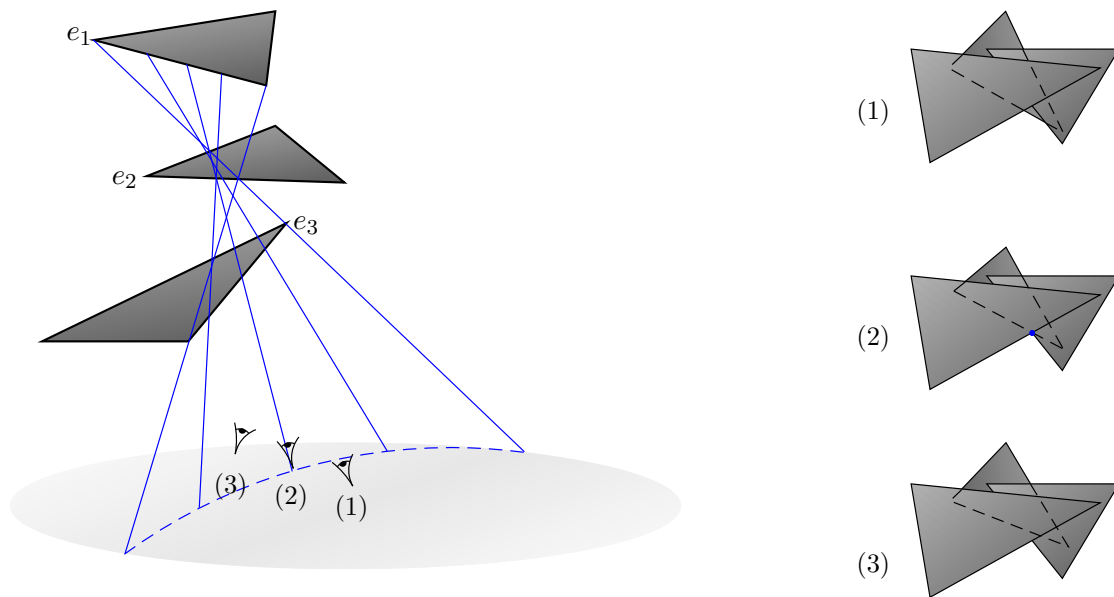


FIG. 1.6 – Un événement visuel EEE (*Edge-Edge-Edge*) est défini par trois arêtes distinctes en positions non dégénérées (ici $e_1e_2e_3$). La surface associée est une quadrique. (1) Au-dessus de cette surface les trois arêtes sont visibles. (2) Au niveau de la surface, elles se superposent (visuellement) en un point. (3) En-dessous, une des trois arêtes disparaît.

d'observation. Les événements visuels interviennent aussi en synthèse d'images : ils correspondent

alors aux contours des ombres dures engendrées par une source ponctuelle, ou délimitent les zones d'ombre, de pénombre et de pleine lumière dans le cas d'une source surfacique. Il apparaît donc essentiel de maîtriser cette information pour obtenir des rendus de qualité.

Les événements visuels ne sont qu'une partie de l'information globale de visibilité. Ils trouvent leurs limites face à des problématiques qui demandent une information complète. Par exemple, les échanges d'énergie entre deux polygones sont corrélés à l'ensemble de leur visibilité. Le problème se pose en illumination globale pour le calcul du facteur de forme, ou encore en télécommunication concernant les échanges d'énergie électromagnétique.

1.1.3 Positionnement

En matière de visibilité, la littérature est des plus vastes. Nous résumons dans cette section les différents critères qui ont pu être proposés pour différencier les problématiques, et permettre leur positionnement dans la littérature. Enfin nous nous appuyons sur ces critères pour situer les travaux présentés dans ce mémoire.

Durand [Dur99] propose un état de l'art multidisciplinaire où les problèmes de visibilité abordés sont classés en fonction de l'espace dans lequel ils sont résolus. Il distingue :

- L'espace image, qui regroupe toutes les méthodes opérant dans un plan projectif 2D. C'est le cas par exemple des approches qui exploitent la carte graphique d'un ordinateur.
- L'espace objet, autrement dit l'environnement 2D ou 3D considéré. En l'occurrence, le classique lancer de rayons entre dans cette catégorie, étant basé sur la recherche de la première intersection rayon/objet.
- L'espace des points de vue, c'est-à-dire l'ensemble de tous les points d'observations possibles dans un environnement donné. Il peut s'agir d'approches liées à la vision par ordinateur, mais cela concerne également les applications type "promenade dans un environnement virtuel".
- L'espace de droites, qui comprend l'ensemble des méthodes analytiques utilisant une paramétrisation des droites. Les travaux en visibilité globale entrent dans cette catégorie.

Nirenstein [NBG02] distingue les différentes méthodes en fonction de la précision avec laquelle elles résolvent un problème de visibilité :

- Les méthodes agressives, qui sous-estiment l'ensemble des visibilité à calculer. Autrement dit, des objets visibles peuvent être manquant, mais il n'existe pas de fausse visibilité.
- Les méthodes conservatrices, qui surestiment l'ensemble des visibilité à calculer. Certaines visibilité rapportées peuvent ne pas exister, mais il est certain que toutes les visibilité existantes sont trouvées.
- Les méthodes exactes, qui trouvent tous les objets visibles, ni plus, ni moins.

La capacité d'une méthode agressive ou conservatrice à tendre vers la solution exacte détermine son efficacité. Il est aussi parfois question de méthode approximative lorsque l'approche suivie ne permet pas de garantir un comportement conservatif ou agressif. Ces critères s'appliquent davantage à des applications dont l'objet est de prouver la visibilité. Ils sont souvent utilisés dans un contexte de calcul de PVS (*Potentially Visible Set*), i.e. de calcul des objets visibles depuis une région de l'espace de vue. Concernant la visibilité globale, un comportement conservatif ou agressif n'a pas de sens.

Bittner [BW03] propose une approche plus générale. Puisque tout ensemble de visibilité peut être asso-

cié à un ensemble de droites, la dimension de cet ensemble constitue un critère commun à tous les problèmes de visibilité. De plus, cette dimension est également une indication sur la difficulté du problème à résoudre. Bittner propose six catégories en fonction des objets impliqués dans le calcul de visibilité. Ces catégories couvrent l'ensemble des problématiques courantes en informatique graphique. Nous les récapitulons ici pour des environnements 3D.

- Visibilité le long d'une droite (0D) (lancer de rayons)
- Visibilité depuis un point (2D)
- Visibilité depuis un segment (3D)
- Visibilité depuis un polygone (4D)
- Visibilité depuis une région (4D)
- Visibilité globale (4D)

Tous ces critères couvrent les différents aspects d'un problème de visibilité. Il n'est pas question ici d'en proposer de nouveaux, mais de préciser l'ordre dans lequel nous les considérons afin de positionner notre problématique.

La première question qui se pose est celle de la dimension du problème auquel on se trouve confronté. Autrement dit, il s'agit de connaître la dimension de l'ensemble de droites associées aux visibilités recherchées. Répondre à cette question, c'est établir la difficulté intrinsèque du problème. Ensuite, il faut distinguer visibilité et visibilité globale.

S'il s'agit de prouver la visibilité d'un ensemble d'objets, alors la précision requise peut fortement simplifier la résolution du problème. Par exemple, prouver agressivement la visibilité de 2 polygones par lancer de rayons permet de transformer un problème 4D par nature en un ensemble de problèmes 0D. En outre, la précision peut conduire à écarter le choix d'un domaine de résolution. Celui-ci est donc à considérer en dernier. Par exemple, les méthodes opérant dans l'espace image, en raison du phénomène de discrétisation, ne peuvent donner une preuve exacte de la visibilité depuis un point de vue.

S'il s'agit de visibilité globale, la question de la précision ne se pose pas. Quant à l'espace de résolution, les travaux en ce domaine s'appuient tous sur une représentation dans un espace dual (de droites ou de rayons).

Les travaux présentés dans ce mémoire concerne le calcul, la représentation et l'exploitation de la visibilité de deux polygones A et B . Pour reprendre les notations de ce début de chapitre, ce qui nous intéresse est l'ensemble 4D suivant :

$$stab(A, B) - \bigcup_{i=1}^n stab(O_i)$$

$\{O_i, 1 \leq i \leq n\}$ étant les bloqueurs pour A et B . Il ne s'agit donc pas seulement de prouver la visibilité ou l'invisibilité de deux polygones. À proprement parler, il ne s'agit pas non plus de visibilité globale, dans la mesure où l'on se restreint à un couple de polygones. Il est plus juste de parler d'une représentation complète des visibilités de deux polygones. Nous verrons cependant que ces trois problématiques peuvent se rejoindre. Par exemple, il n'existe pas à ce jour de solution exacte permettant de prouver la visibilité de deux polygones sans devoir, le cas échéant, calculer l'ensemble des visibilités et montrer que celui-ci est vide ou non vide. Nos travaux s'inscrivent dans la continuité de ceux de Bittner [Bit02] et Nirenstein [Nir03] que nous présenterons dans la suite de ce chapitre.

1.2 Prérequis mathématiques

Cette partie présente les principales notions de géométrie sous-jacentes à notre problématique. Après quelques notions de géométrie projective, nous présentons l'espace de Plücker utilisé pour paramétrer les droites de \mathbb{R}^3 . Ensuite, plusieurs rappels seront faits sur les notions de polyèdres et de polytopes. Enfin, nous verrons comment ces outils mathématiques permettent d'apporter des solutions au problème de la visibilité entre des polygones de \mathbb{R}^3 .

1.2.1 Géométrie projective

La géométrie projective présente l'avantage de modéliser les notions intuitives de perspective et d'horizon. De fait, son usage est très répandu en vision par ordinateur. Prenons une image simple : les rails d'un chemin de fer sont parallèles et jamais ne s'intersectent. Mais, par perspective, un observateur verra ces mêmes rails s'intersecter à l'horizon. Contrairement à la géométrie "classique", la géométrie projective permet de prendre en compte une position d'observation.

Cette section donne les éléments de géométrie projective utiles à la suite de ce chapitre. Pour une présentation approfondie et rigoureuse, nous renvoyons le lecteur au livre de Jorge Stolfi [Sto91].

Géométrie projective classique

Un espace projectif se définit pour toute dimension de la manière suivante :

Définition 7 *Étant donné un corps \mathbb{K} , un espace projectif de dimension n sur \mathbb{K} , noté $\mathbb{P}^n(\mathbb{K})$ ou $\mathbb{K}\mathbb{P}^n$, est l'ensemble des droites vectorielles de \mathbb{K}^{n+1} . Autrement dit, l'espace projectif $\mathbb{P}^n(\mathbb{K})$ est l'ensemble des classes d'équivalence induites par la relation :*

$$x \sim \lambda x, x \in \mathbb{K}^{n+1} \setminus \{0\}, \lambda \in \mathbb{K} \setminus \{0\}$$

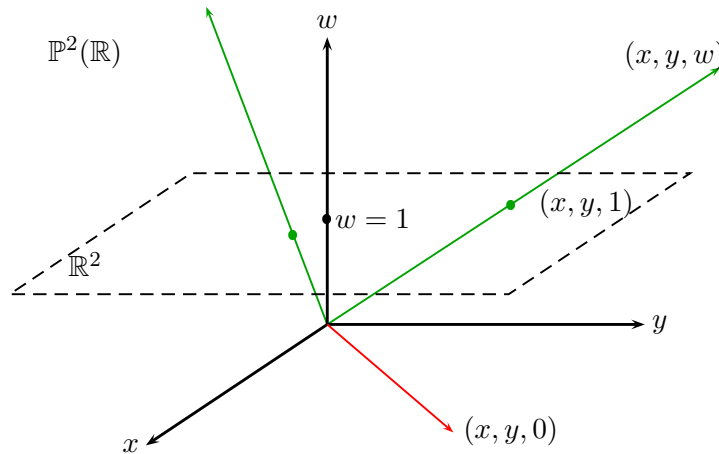


FIG. 1.7 – Illustration du plan projectif $\mathbb{P}^2(\mathbb{R})$. Tout point de \mathbb{R}^2 correspond à un point de $\mathbb{P}^2(\mathbb{R})$ (i.e. une droite vectorielle de \mathbb{R}^3). Notons que, relativement à l'hyperplan de projection choisi (ici $w = 1$), les points de type $\langle x, y, 0 \rangle$ de $\mathbb{P}^2(\mathbb{R})$ n'ont pas d'équivalent dans \mathbb{R}^2 . Ce sont les points à l'infini, qui existent uniquement dans le plan projectif.

Un point $x \in \mathbb{K}^{n+1} \setminus \{0\}$ détermine une classe d'équivalence $\langle x_0, \dots, x_n \rangle$ (i.e. la droite passant par x dans $\mathbb{P}^n(\mathbb{K})$) de manière unique à une constante multiplicative non nulle près. On parle alors de *coordonnées projectives* ou *coordonnées homogènes* des points d'un espace projectif.

Par la suite, nous verrons que les travaux repris dans ce mémoire passent d'un système de coordonnées homogènes à un système de coordonnées cartésien en choisissant un *hyperplan de projection*. Par exemple, considérons l'hyperplan de projection $x_0 = 1$ dans \mathbb{K}^{n+1} . Tout point de $\mathbb{P}^n(\mathbb{K})$ (i.e. une droite vectorielle dans \mathbb{K}^{n+1}), non parallèle à $x_0 = 1$ intersecte l'hyperplan de projection en $(1, \frac{x_1}{x_0}, \dots, \frac{x_n}{x_0})$ (ce qui revient à fixer un unique représentant par classe d'équivalence). Ainsi, d'un ensemble de classes d'équivalence $\langle x_0, \dots, x_n \rangle$, on passe à un ensemble de points $(1, \frac{x_1}{x_0}, \dots, \frac{x_n}{x_0})$ de \mathbb{K}^{n+1} , ou indifféremment, à l'ensemble des points $(\frac{x_1}{x_0}, \dots, \frac{x_n}{x_0})$ de \mathbb{K}^n . On se ramène ainsi à un système de coordonnées cartésien. Soulignons que ce changement de système ne conserve pas les points de $\mathbb{P}^n(\mathbb{K})$ parallèles à l'hyperplan de projection choisi (dans notre exemple, il s'agit des points $\langle 0, x_1, \dots, x_n \rangle$).

Le principe de dualité est une autre notion importante en géométrie projective. Prenons par exemple deux points a et b dans $\mathbb{P}^n(\mathbb{K})$, ayant respectivement pour coordonnées homogènes (a_0, \dots, a_n) et (b_0, \dots, b_n) , et tel que $a_0 b_0 + \dots + a_n b_n = 0$ soit vérifiée. On peut alors dire soit que le point a est incident à l'hyperplan $b_0 x_0 + \dots + b_n x_n = 0$, soit que le point b est incident à l'hyperplan $a_0 x_0 + \dots + a_n x_n = 0$. Dans un espace projectif, il est toujours possible de définir une correspondance entre les points et les hyperplans tout en préservant les relations d'incidence. De cette propriété découle le principe de dualité suivant :

Définition 8 *Tout théorème ou proposition en géométrie projective possède automatiquement son théorème dual ou sa proposition duale pour lesquels les termes “points” et “hyperplans” ont été échangés dans l'énoncé.*

Ainsi toute démonstration dans un espace projectif permet de prouver simultanément deux théorèmes différents.

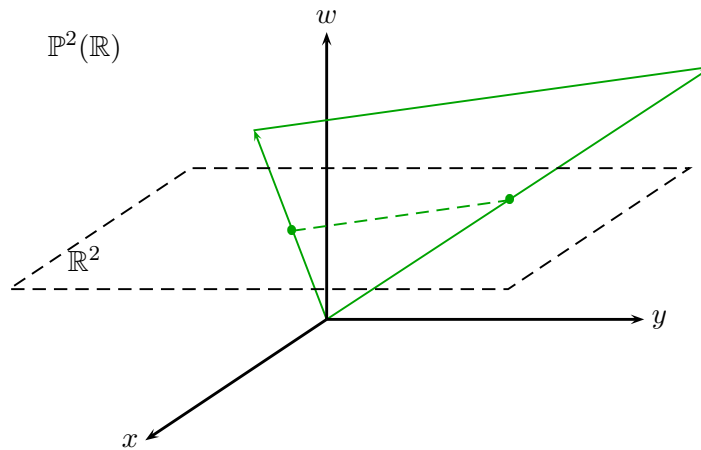


FIG. 1.8 – Une droite de \mathbb{R}^2 correspond à un sous-espace vectoriel de dimension 2 dans $\mathbb{P}^2(\mathbb{R})$, i.e. une droite de $\mathbb{P}^2(\mathbb{R})$ représentable par un plan passant par l'origine dans \mathbb{R}^3 .

Pour illustrer ces définitions, donnons une brève description du plan projectif $\mathbb{P}^2(\mathbb{R})$, (i.e., l'ensemble des droites vectorielles de \mathbb{R}^3). Un point du plan projectif ayant pour coordonnées homogènes (x, y, w) réfère à la classe d'équivalence $\langle x, y, w \rangle$ (i.e. la droite passant par l'origine et le point (x, y, w) dans \mathbb{R}^3). Choisissons arbitrairement le plan de projection $w = 1$, tout point (x, y) de \mathbb{R}^2 est en bijection avec le point $(x, y, 1)$ du plan de projection choisi, et chaque point $(x, y, 1)$ correspond dans $\mathbb{P}^2(\mathbb{R})$ à la

classe d'équivalence $\langle x, y, 1 \rangle$. Ceci illustre que tout point de \mathbb{R}^2 est représentable dans $\mathbb{P}^2(\mathbb{R})$. En revanche, la réciproque est fausse. Vis à vis du plan de projection $w = 1$ choisi, les classes d'équivalence de type $\langle x, y, 0 \rangle$ (i.e., les droites vectorielles du plan $w = 0$) n'ont pas d'équivalent dans \mathbb{R}^2 . Elles correspondent aux points à l'infini. En ce sens, le plan projectif peut être vu comme le plan classique \mathbb{R}^2 augmenté d'une droite à l'infini. La figure 1.7 résume ce propos.

Un avantage du système de coordonnées homogènes est de s'affranchir de tous les cas particuliers de la géométrie cartésienne. Considérons, par exemple, le cas d'intersection de deux droites. Si un point dans \mathbb{R}^2 correspond à un point de $\mathbb{P}^2(\mathbb{R})$ (i.e. une droite vectorielle de \mathbb{R}^3), une droite dans \mathbb{R}^2 correspond à une droite dans $\mathbb{P}^2(\mathbb{R})$ (i.e. un plan passant par l'origine dans \mathbb{R}^3). La figure 1.8 donne une illustration. Aussi, l'intersection de deux droites dans \mathbb{R}^2 se traduit par l'intersection de deux droites dans le plan projectif (i.e. l'intersection de deux plans passant par l'origine dans \mathbb{R}^3). Une singularité dans \mathbb{R}^2 est que deux droites parallèles ne s'intersectent jamais. Dans le plan projectif (en coordonnées homogènes) toutes les droites s'intersectent. Si elles sont parallèles, alors leur intersection a lieu en un point à l'infini. La figure 1.9 donne un exemple pour des droites sécantes et parallèles.

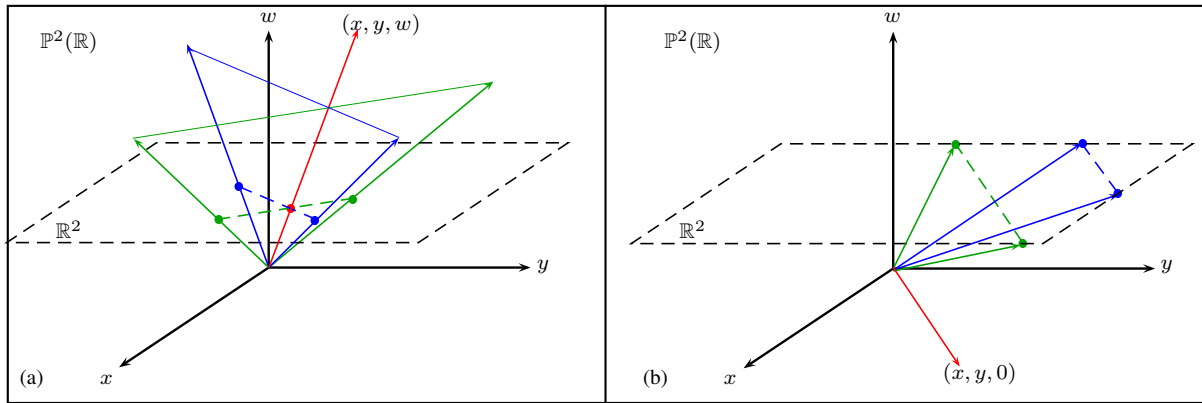


FIG. 1.9 – (a) : l'intersection de deux droites dans \mathbb{R}^2 correspond toujours à l'intersection de deux droites dans $\mathbb{P}^2(\mathbb{R})$ (i.e., l'intersection de deux plans de \mathbb{R}^3 passant par l'origine). (b) si deux droites parallèles ne s'intersectent pas dans \mathbb{R}^2 , elles s'intersectent dans $\mathbb{P}^2(\mathbb{R})$ en un point à l'infini, ici de type $\langle x, y, 1 \rangle$ (deux plans de \mathbb{R}^3 passant par l'origine s'intersectent toujours en une droite).

L'exemple du plan projectif illustre les principes et avantages offerts par la géométrie projective et son système de coordonnées homogènes. En particulier, l'absence de singularité simplifie l'analyse de problèmes géométriques en évitant de traiter séparément les cas particuliers. De fait, le développement d'algorithmes est également facilité.

Géométrie projective orientée

La géométrie projective classique ne permet pas de distinguer l'orientation de primitives géométriques. C'est un inconvénient majeur dans la mesure où il est courant de travailler avec des plans, des triangles ou encore des droites possédant une orientation. La géométrie projective orientée permet de pallier ce problème.

De manière imagée, un espace projectif orienté est une double copie d'un espace projectif classique, chaque copie possédant une orientation opposée. Cela revient à distinguer pour chaque classe d'équivalence $\langle x_0, \dots, x_n \rangle$ les classes $\langle -x_0, \dots, -x_n \rangle$ et $\langle +x_0, \dots, +x_n \rangle$. Les espaces projectifs

orientés permettent ainsi de tenir compte de l'orientation des primitives géométriques tout en bénéficiant des mêmes avantages que les espaces projectifs classiques.

1.2.2 Paramétrisation de Plücker

L'espace de Plücker [Som59, Sto91] est un espace projectif orienté de dimension 5 $\mathbb{P}^5(\mathbb{R})$, que nous noterons \mathbb{P}^5 . Il permet une paramétrisation sans singularité des droites réelles orientées de \mathbb{R}^3 . Les coordonnées de Plücker sont parfois présentées comme un cas particulier des coordonnées de Grassmann. Considérant un sous espace affine \mathcal{W} de dimension k contenu dans un espace \mathcal{V} de dimension n , les coordonnées de Grassmann permettent de paramétrer \mathcal{W} par un point dans un espace projectif de dimension $\binom{n+1}{k+1} - 1$. Dans le cas particulier de la paramétrisation des droites de \mathbb{R}^3 , $k = 1$ et $n = 3$. Considérons une droite l passant par le point (p_x, p_y, p_z) puis par le point (q_x, q_y, q_z) . Elle est définie dans l'espace de Plücker par le point l^* $(\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5)$ tel que :

$$\begin{array}{lll} \pi_0 = q_x - p_x & \pi_1 = q_y - p_y & \pi_2 = q_z - p_z \\ \pi_3 = q_z p_y - q_y p_z & \pi_4 = q_x p_z - q_z p_x & \pi_5 = q_y p_x - q_x p_y \end{array}$$

La paramétrisation de Plücker n'est pas intuitive. Les trois premiers paramètres correspondent au vecteur directeur de la droite. Les trois derniers encodent la position de la droite dans l'espace et correspondent à un vecteur orthogonal au plan formé par la droite et l'origine du repère. Cette interprétation est résumée sur la figure 1.10.

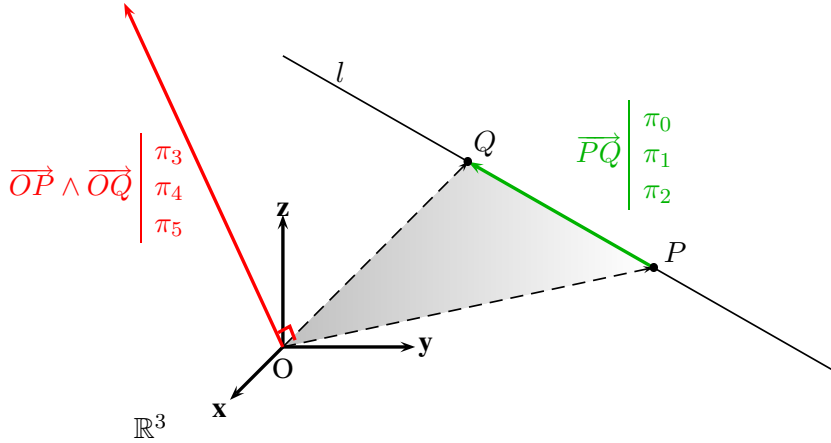


FIG. 1.10 – Interprétation des coordonnées de Plücker : les coordonnées de Plücker $(\pi_0, \pi_1, \pi_2, \pi_3, \pi_4, \pi_5)$ de la droite l encodent la direction de l sur les trois premières coordonnées et sa position dans l'espace sur les trois dernières. (π_0, π_1, π_2) correspond au vecteur directeur de la droite. (π_3, π_4, π_5) est le vecteur orthogonal au plan (OPQ) passant par l et l'origine du repère.

Tout point π de l'espace de Plücker peut être associé à un hyperplan dual h_π défini par :

$$h_\pi = \{x \in \mathbb{P}^5 \mid \pi_3 x_0 + \pi_4 x_1 + \pi_5 x_2 + \pi_0 x_3 + \pi_1 x_4 + \pi_2 x_5 = 0\}$$

Par conséquent, il y a deux façons de considérer une droite dans l'espace de Plücker : soit par un point, soit par son hyperplan associé. Cependant, tout point de \mathbb{P}^5 (ou tout hyperplan) ne correspond pas nécessairement à une droite orientée dans \mathbb{R}^3 . L'ensemble des droites réelles forme dans l'espace de Plücker une quadrique 4D souvent appelée *hypersurface de Plücker* (parfois aussi *quadrique de Klein* ou encore *variété de Grassmann*). Elle est définie par :

$$\mathcal{Q} = \{x \in \mathbb{P}^5 \mid h_x(x) = 0\} \setminus \{0\}$$

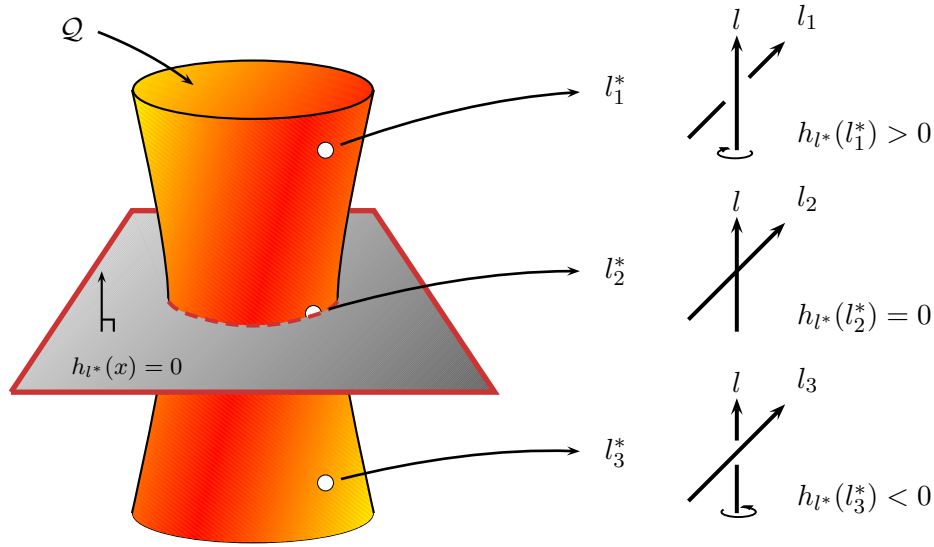


FIG. 1.11 – Orientation relative de deux droites : l’hyperplan h_{l^*} partitionne les points sur la quadrique de Plücker Q et induit trois ensembles de droites orientées. Les droites qui tournent autour de l dans le sens direct, les droites incidentes ou parallèles à l , et les droites qui tournent autour de l dans le sens indirect.

Seuls les points de la quadrique de Plücker correspondent à une droite orientée de \mathbb{R}^3 . Les autres points sont parfois dits correspondre à des droites imaginaires. La paramétrisation de Plücker permet de caractériser facilement l’orientation relative de deux droites, c’est-à-dire dans quel sens une droite “tourne” autour d’une autre. Pour donner une image, imaginons une girouette : selon la direction du vent, la girouette tourne dans le sens des aiguilles d’une montre, ou bien dans le sens contraire. Pour illustrer notre propos, il suffit d’assimiler la direction du vent ainsi que l’axe de la girouette à deux droites orientées. Considérons une droite l et son hyperplan associé h_{l^*} dans l’espace de Plücker. Ce dernier induit un demi-espace négatif $h_{l^*}^-$ et un demi-espace positif $h_{l^*}^+$. Une droite l_2 est incidente ou parallèle à l_1 si son point de Plücker l_2^* est incident à h_{l^*} . Si tel n’est pas le cas, l’appartenance de l_2^* à $h_{l^*}^-$ ou $h_{l^*}^+$ définit l’orientation relative de la droite l_2 par rapport à la droite l . Cette propriété est illustrée par la figure 1.11. Il suffit donc d’évaluer le signe de $h_{l^*}(l_2^*)$ dans l’espace de Plücker pour déterminer l’orientation relative des droites associées.

Tout comme pour les coordonnées d’un point, il est possible d’interpréter géométriquement dans \mathbb{R}^3 la quantité $h_{l^*}(l_2^*)$. Comme indiqué sur la figure 1.12, elle est proportionnelle au volume signé d’un tétraèdre défini par les 4 points associés aux deux droites considérées.

La paramétrisation de Plücker et les propriétés présentées ici constituent un outil puissant. Elles permettent en particulier une représentation aisée d’ensembles de droites par des polyèdres convexes, ou des polytopes.

1.2.3 Polyèdres et polytopes

Pour la compréhension de la suite de ce chapitre, il est nécessaire de rappeler certaines définitions sur les polyèdres. Les définitions proposées reprennent celles de [HJGZ97]. Il existe différentes manières de définir un polyèdre convexe dans \mathbb{R}^d . Nous retiendrons la définition suivante :

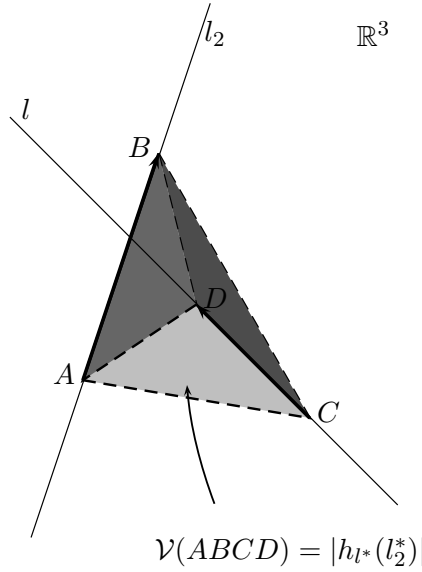


FIG. 1.12 – Interprétation géométrique de l'évaluation d'un point par rapport à un hyperplan dans l'espace de Plücker : $l = (AB)$ et $l_2 = (CD)$ étant deux droites non incidentes, $h_{l^*}(l_2^*)$ est proportionnelle au volume signé du tétraèdre $(ABCD)$. Si les deux droites étaient incidentes, le tétraèdre associé serait dégénéré et son volume nul.

Définition 9 Un polyèdre convexe est l'ensemble des points $x \in \mathbb{R}^d$ satisfaisant un système de n inéquations linéaires :

$$mx \leq b$$

Où m est une matrice de $\mathbb{R}^{n \times d}$ et b un élément de \mathbb{R}^n .

De manière équivalente, un polyèdre convexe peut être considéré comme l'intersection de n demi-espaces dans \mathbb{R}^d . En effet, notant m_i , $1 \leq i \leq n$ chaque ligne de la matrice m , et b_i la composante d'indice i du vecteur $b \in \mathbb{R}^d$, $m_i x - b_i = 0$, définit l'équation de hyperplan H_i . Dans ce cas, l'ensemble solution du système précédent est l'ensemble des points $x \in \mathbb{R}^d$ tels que :

$$x \in \bigcap_{i=1}^n H_i^-$$

Définition 10 Un polytope est un polyèdre convexe et fermé.

La définition 9 n'implique pas qu'un polyèdre convexe soit fermé. Lorsque tel est le cas, on parle alors de polytope (définition 10). Il existe deux façons de représenter un polytope. La première consiste à le définir en tant que solution d'un système fini d'inéquations linéaires (de manière analogue à la définition 9 d'un polyèdre). On parle alors de H-représentation (*Hyperplane representation*) ou encore de H-polytope. La seconde possibilité est de considérer le polytope comme l'enveloppe convexe d'un ensemble fini $V = \{v^1, \dots, v^k\}$ de k points de \mathbb{R}^d . Autrement dit c'est l'ensemble des points définis par :

$$\text{conv}(V) = \left\{ \sum_{j=1}^k \lambda_j v^j \mid \lambda_j \in \mathbb{R}^+, \sum_{j=1}^k \lambda_j = 1 \right\}$$

V est l'ensemble des sommets du polytope. On parle alors de V-représentation (*Vertex representation*) ou encore de V-polytope. La figure 1.13 illustre ces deux représentations possibles d'un polytope.

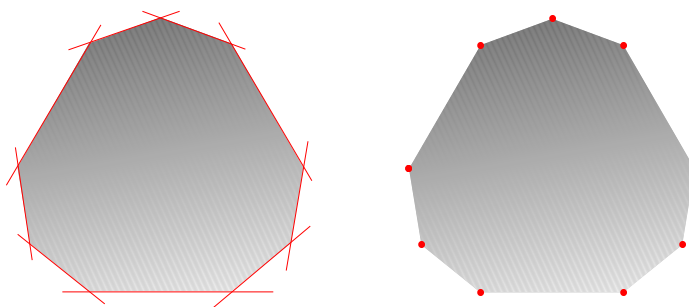


FIG. 1.13 – A gauche, représentation par hyperplans (H-représentation) d'un polytope de \mathbb{R}^2 . A droite, sa représentation par sommets (V-représentation)

Ces deux représentations sont bien entendu équivalentes. Le passage d'une représentation à une autre correspond à deux problèmes algorithmiques bien connus. L'*énumération des sommets* d'un polytope consiste à obtenir sa V-représentation à partir de sa H-représentation. Inversement, l'*énumération des facettes* consiste à calculer la H-représentation du polytope à partir de sa V-représentation (un cas particulier du problème classique de calcul d'enveloppe convexe).

Définition 11 *Un d -polytope est un polytope de dimension d .*

La dimension de l'enveloppe convexe de la V-représentation d'un polytope définit sa dimension. On note alors d -polytope un polytope de dimension d (définition 11).

Définition 12 *Le bord d'un d -polytope \mathcal{P} de \mathbb{R}^d est constitué de k -polytopes, $0 \leq k < d$, appelés faces ou k -faces.*

Le bord (définition 12) d'un d -polytope est formé de faces de dimensions inférieures. En particulier, les 0-faces sont les sommets du polytope, les 1-faces ses arêtes. Les $(d - 1)$ -faces sont aussi appelées facettes. Une facette est contenue par exactement 1 hyperplan de la H-représentation du polytope. Plus généralement, une k -face est définie par l'intersection de $(d - k)$ hyperplans. Cet ensemble d'hyperplan forme la description combinatoire de la k -face :

Définition 13 *La description combinatoire d'une k -face est la liste des hyperplans qui la contiennent.*

Définition 14 *Le graphe d'incidence d'un polytope est l'ensemble des faces du polytope ordonnées au sens de l'inclusion.*

Le graphe d'incidence (définition 14) d'un polytope est une représentation ordonnée de ses faces. Chaque nœud du graphe est associé à une face du polytope. Un arc relie une $(k + 1)$ -face à une k -face si la première contient la seconde. Par extension, le polytope lui-même est considéré comme une face contenant toutes les autres. La figure 1.14 donne un exemple avec le graphe d'incidence d'un triangle dans le plan. Un graphe d'incidence peut être calculé pour l'ensemble des faces au plus égales à k . On parle alors du k -squelette d'un polytope.

Définition 15 *Un d -polytope est dit simple si chaque sommet de sa V-représentation est contenu exactement dans d facettes.*

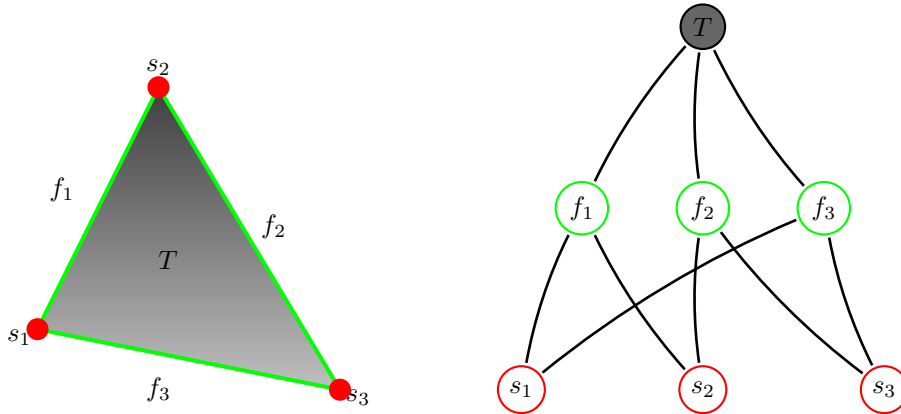


FIG. 1.14 – Le graphe d'incidence d'un polytope de \mathbb{R}^n représente les relations d'incidence entre ses faces de dimension k , $0 \leq k \leq n$. Ici, le graphe d'incidence d'un triangle T (un polytope de \mathbb{R}^2). $\{s_1, s_2, s_3\}$ sont ses 0-faces, $\{f_1, f_2, f_3\}$ ses 1-faces. T est représenté par une 2-face contenant toutes les faces de dimensions inférieures. Par extension un polytope de \mathbb{R}^d est toujours représenté par une unique d -face

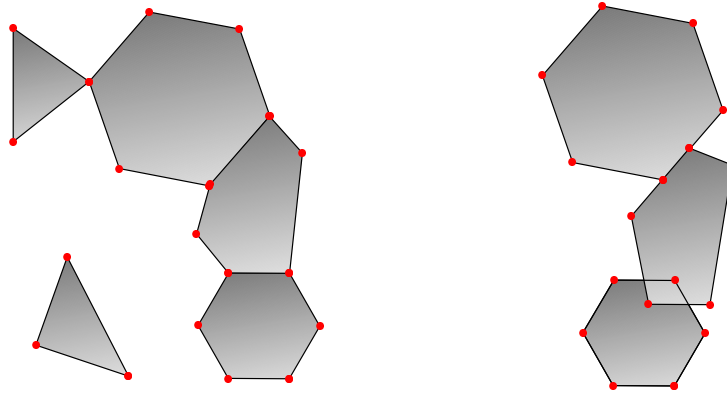


FIG. 1.15 – A gauche, un complexe de polytopes de \mathbb{R}^2 . L'intersection de deux polytopes est vide ou est égale à une face commune aux deux polytopes. A droite l'ensemble des 3 polytopes ne forment pas un complexe, leurs intersections ne correspondant à aucune face.

Lorsque chaque sommet d'un d -polytope est incident à exactement d hyperplans de sa H -représentation, le polytope est dit simple (définition 15). Cette propriété permet notamment de résoudre plus efficacement les problèmes d'énumération de sommets ou facettes.

Définition 16 Un complexe de polytopes de \mathbb{R}^d est un ensemble de polytopes de dimension d vérifiant les deux conditions suivantes : chaque k -face de polytope est élément du complexe, et si deux faces s'intersectent, leur intersection est une face commune.

La définition 16 introduit la notion de “complexe de polytopes”. La figure 1.15 en donne un exemple et un contre exemple. Par la suite, nous utiliserons parfois ce terme abusivement dans le cas où l'intersection de deux k -faces n'est pas une face commune, mais seulement contenue dans une face commune. On parle alors parfois de complexe faible de polytopes.

1.3 Visibilité et espace de Plücker

Cette section montre comment la paramétrisation de Plücker permet de formuler des solutions à deux problèmes fondamentaux de visibilité :

1. Etant donné un ensemble $\{A_1, \dots, A_n\}$ de polygones convexes dans \mathbb{R}^3 , calculer une représentation de $stab(A_1, \dots, A_n)$.
2. Etant donnés deux ensembles $\{A_1, \dots, A_n\}$ et $\{O_1, \dots, O_m\}$ de polygones convexes dans \mathbb{R}^3 , calculer une représentation de $stab(A_1, \dots, A_n) - \cup_{j=1}^m stab(O_j)$.

Le premier problème est un cas particulier du second (il suffit de considérer $m = 0$). Il concerne la visibilité à travers une séquence d'ouvertures convexes. Il intervient par exemple dans des calculs d'illumination pour des intérieurs : les ouvertures correspondent alors aux portes séparant les pièces de la scène. L'enjeu est alors de déterminer les couples de pièces qui peuvent échanger de l'énergie lumineuse à travers une série de portes [Tel92b, TH93].

Le second problème fait intervenir un ensemble de bloqueurs $\{O_1, \dots, O_m\}$. Pour reprendre l'exemple du calcul de l'illumination d'un intérieur, cela revient à prendre désormais en compte le mobilier contenu dans les pièces et qui peut obstruer, totalement ou partiellement, la visibilité de deux pièces à travers une séquence de portes. En particulier, pour $n = 2$, cela équivaut au calcul de la représentation des visibilités de deux polygones, *i.e.* la problématique qui nous intéresse.

Après avoir expliqué les solutions à ces deux problèmes dans l'espace de Plücker, nous montrons l'interprétation qui peut en être faite en terme d'ensemble de droites critiques et d'événements visuels.

1.3.1 Visibilité à travers une séquence de polygones

Le premier problème abordé ici est la représentation des droites intersectant tous les polygones d'un ensemble donné. Teller [TH91] est le premier à apporter une solution pratique à cette question. Il utilise la paramétrisation de Plücker afin de caractériser l'ensemble de droites recherché.

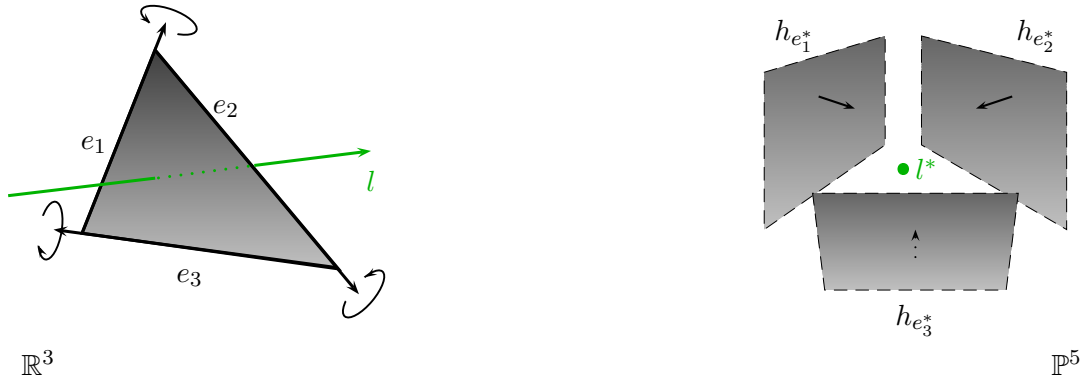


FIG. 1.16 – Caractérisation des droites intersectant un triangle : les arêtes e_1, e_2, e_3 du triangle sont orientées de manière consistante. Elles induisent dans l'espace de Plücker 3 hyperplans $h_{e_1}^*, h_{e_2}^*$ et $h_{e_3}^*$. Toute droite l qui intersecte le triangle a son point de Plücker dans l'intersection des 3 demi-espaces positifs $h_{e_1}^+ \cap h_{e_2}^+ \cap h_{e_3}^+$ (ou l'intersection des 3 demi-espaces négatifs selon l'orientation de l).

Dans un premier temps, considérons un unique triangle dont les arêtes e_1, e_2 et e_3 sont orientées de manière consistante. Les droites supports à ces 3 arêtes induisent 3 hyperplans dans l'espace de Plücker. Toute droite orientée l intersectant le triangle a une orientation consistante vis-à-vis de chaque arête du triangle. Sans restriction aucune, nous pouvons supposer que cette orientation est positive (sinon il suffit

d'inverser l'orientation des arêtes du triangle). Ce qui signifie dans l'espace de Plücker que le point l^* est contenu dans l'intersection des 3 demi-espaces positifs $h_{e_1^+}^+, h_{e_2^+}^+, h_{e_3^+}^+$. C'est ce que montre la figure 1.16. Ainsi les hyperplans associés aux arêtes du triangle permettent de caractériser ses droites poignardantes par l'intersection de demi-espaces dans l'espace de Plücker.

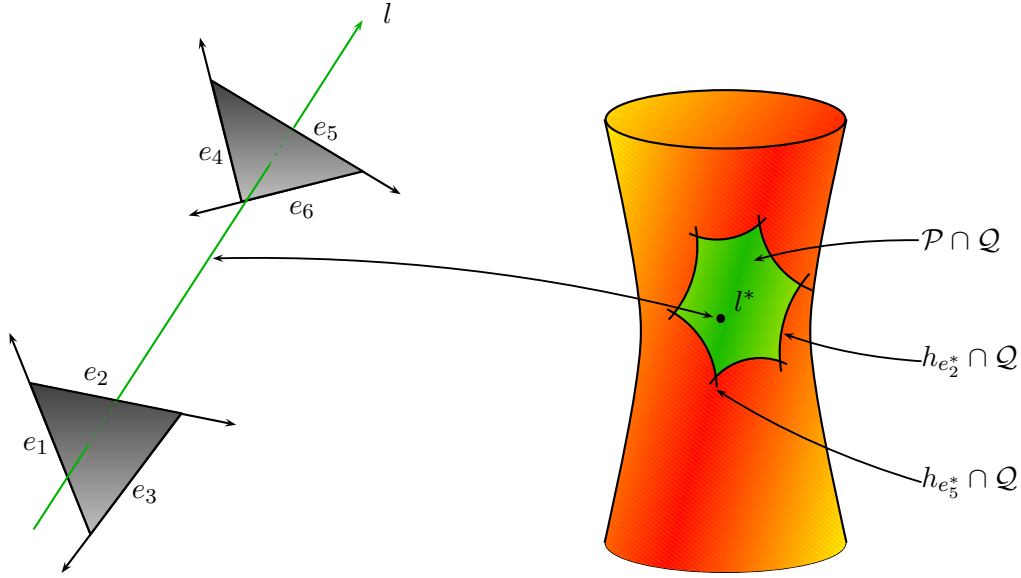


FIG. 1.17 – Les hyperplans $h_{e_i^*}$ associés aux arêtes e_i des polygones forment un polyèdre convexe \mathcal{P} dans \mathbb{P}^5 . L'intersection de \mathcal{P} avec la quadrique de Plücker \mathcal{Q} est exactement l'ensemble des droites intersectant les polygones (sur le schéma, c'est l'intersection de \mathcal{P} avec la quadrique \mathcal{Q} qui est représentée).

Ceci se généralise directement aux droites intersectant un ensemble de polygones convexes. Ayant choisi une orientation commune pour les arêtes e_i ($1 \leq i \leq n$) de tous les polygones, toute droite l poignardant chacun des polygones vérifie :

$$l^* \in \bigcap_{j=1}^n h_{e_j^*}^+$$

En particulier si l'intersection des demi-espaces et de la quadrique de Plücker est vide, il n'existe pas de droite pouvant intersecter tous les polygones de l'ensemble. L'intersection des demi-espaces répond dans \mathbb{P}^5 à la définition d'un polyèdre convexe \mathcal{P} . Teller note que si ce polyèdre n'est pas nécessairement fermé, son intersection avec la quadrique de Plücker l'est obligatoirement. En effet, à l'image de la baguette du joueur de triangle, une droite poignardante que l'on pourrait déplacer finira toujours contrainte par les arêtes des polygones qu'elle intersecte. L'ensemble des droites réelles défini par l'intersection du polyèdre et la quadrique de Plücker est ainsi exactement l'ensemble des droites poignardantes. La figure 1.17 illustre la construction pour deux triangles.

En pratique, il est préférable de manipuler un polytope plutôt qu'un polyèdre non borné. D'une part, il est plus simple de travailler avec une représentation fermée d'un point de vue algorithmique. D'autre part, cela permet de restreindre les calculs à la zone de la quadrique de Plücker qui seule possède une équivalence réelle. Ainsi des hyperplans supplémentaires peuvent être ajoutés afin de borner le polyèdre sans affecter son intersection avec la quadrique.

1.3.2 Prise en compte de l'occultation

Considérons maintenant l'évolution de la problématique précédente. Comment déterminer toutes les droites intersectant un ensemble $\{A_1, \dots, A_n\}$ de polygones sans intersecter ceux d'un ensemble de bloqueurs $\{O_1, \dots, O_m\}$? En 2002, Nirenstein [NBG02] publie la première solution pratique à ce problème. Peu de temps après Bittner [Bit02] propose dans sa thèse une approche sensiblement différente. Dans les deux cas, ces travaux utilisent la paramétrisation de Plücker pour résoudre le problème en effectuant des opérations de type CSG (*Constructive Solid Geometry*) sur des polytopes. Ce sont ces principes communs que nous présentons dans cette section. A la fin de ce chapitre, nous reviendrons en détail sur les approches respectives de ces auteurs.

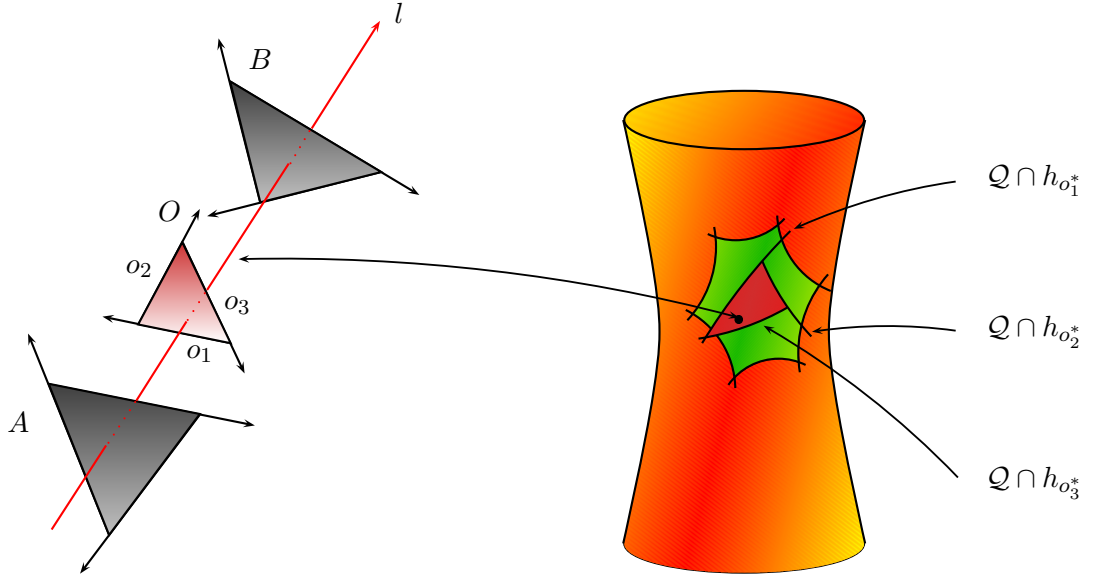


FIG. 1.18 – Caractérisation des droites intersectant les triangles A et B sans intersecter O . Le polyèdre convexe représentatif des droites poignardantes de O est soustrait au polytope représentatif des droites poignardant A et B . Ce dernier est donc successivement intersecté par les hyperplans de la H-représentation du polyèdre O . Puis la partie commune au polytope et au polyèdre peut être supprimée. Le résultat est un complexe de polytopes dont l'intersection avec la quadrique \mathcal{Q} est l'ensemble des droites intersectant A et B mais pas O (pour plus de clarté, seules les intersections avec la quadrique des polytopes ou polyèdres sont représentées).

Notons \mathcal{P}_A le polytope représentant toutes les droites intersectant les polygones de $\{A_1, \dots, A_n\}$. \mathcal{P}_{O_j} ($1 \leq j \leq m$) désigne le polyèdre associé au bloqueur d'indice j dans $\{O_1, \dots, O_m\}$. L'ensemble des droites intersectant $\{A_1, \dots, A_n\}$ mais pas $\{O_1, \dots, O_m\}$ est :

$$\mathcal{P}_A - \bigcup_{j=1}^m \mathcal{P}_{O_j}$$

Le problème se résout en supprimant successivement les parties communes à \mathcal{P}_A et à chaque \mathcal{P}_{O_j} . La soustraction successive des bloqueurs engendre un ensemble de polytopes formant un complexe. Les hyperplans de la H-représentation de chaque bloqueur permettent d'intersecter le complexe et d'éliminer la partie contenue dans le polyèdre. Lorsque tous les bloqueurs sont soustraits, l'intersection du complexe résultant avec la quadrique est exactement l'ensemble des droites intersectant $\{A_1, \dots, A_n\}$ sans intersecter $\{O_1, \dots, O_m\}$. La figure 1.18 illustre le procédé avec 2 polygones et 1 bloqueur. Nous reviendrons dans la section 1.4 sur les solutions algorithmiques existantes pour mettre en œuvre cette approche.

1.3.3 Événements visuels étendus

Comme nous venons de le voir, la paramétrisation de Plücker constitue un outil permettant de représenter les visibilitées de polygones convexes par un ensemble de polytopes. Dans cette section, nous expliquons ce que représentent les différentes k -faces des bords des polytopes. En outre nous précisons ce qui distingue cette approche d'une étude de visibilité globale ainsi que les rapprochements qui peuvent être faits.

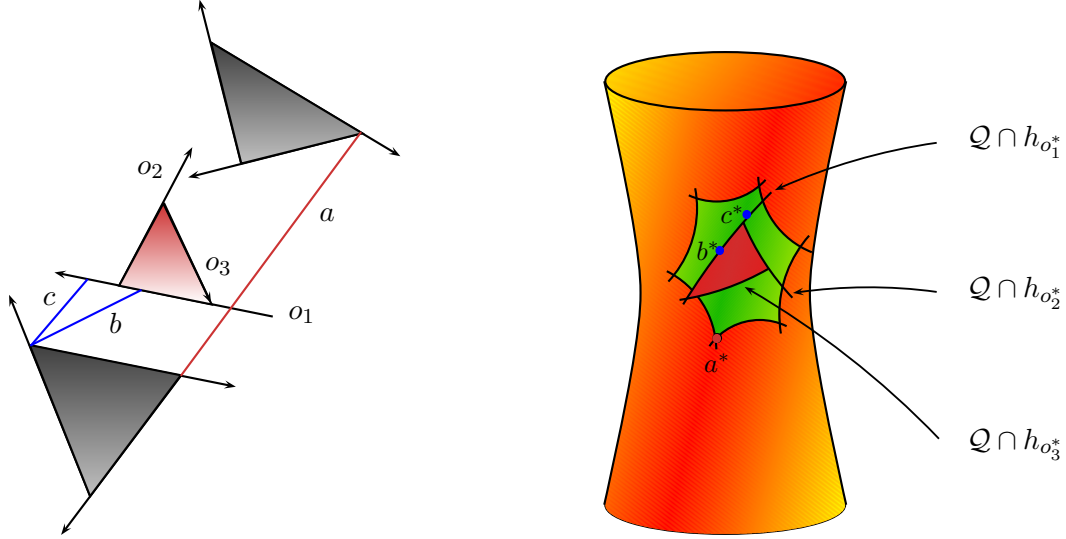


FIG. 1.19 – Les bords d'un complexe de polytopes pour la visibilité de deux polygones contient les événements visuels et les droites critiques au sens de la visibilité globale. Prenons par exemple le cas de la droite sommet à sommet a (ici, notre illustration ne peut faire apparaître que 2 des 4 hyperplans contenant son point de Plücker a^*). Par construction ces ensembles critiques sont prolongés le long des droites support aux arêtes. Ainsi la droite b appartient à un événement VE, ce qui n'est pas le cas de la droite c . Cependant les deux droites appartiennent à la même 2-face (là encore, le schéma en donne une illustration simplifiée puisqu'un seul des 3 hyperplans est indiqué). L'événement est étendu le long de la droite o_1 . Sur le schéma, les droites o_1 , o_2 et o_3 sont prolongées en conséquence.

Par construction, les hyperplans de la H-représentation d'un polytope correspondent à des droites réelles. Une k -face d'un polytope de \mathbb{R}^d est définie par l'intersection (d'au moins) $(d-k)$ hyperplans (ici $d = 5$). Et, si elle existe, l'intersection d'une k -face ($k > 0$) avec la quadrique de Plücker est un ensemble de droites réelles de dimension $k-1$, que l'on désigne aussi par trace ou $k-1$ -trace. Considérant les k -faces possédant une intersection avec la quadrique, il en résulte l'interprétation suivante :

- Une arête (1-face) d'un polytope définie par l'intersection de 4 hyperplans $\{h_{l_1^*} h_{l_2^*} h_{l_3^*} h_{l_4^*}\}$ représente une droite réelle (0-trace) incidente aux 4 droites $\{l_1 l_2 l_3 l_4\}$.
- Une 2-face d'un polytope définie par l'intersection de 3 hyperplans $\{h_{l_1^*} h_{l_2^*} h_{l_3^*}\}$ représente un ensemble de droites réelles de dimension 1 (1-trace) incidentes aux droites $\{l_1 l_2 l_3\}$.
- Une 3-face d'un polytope définie par l'intersection de 2 hyperplans $\{h_{l_1^*} h_{l_2^*}\}$ représente un ensemble de droites réelles de dimension 2 (2-trace) incidentes aux droites $\{l_1 l_2\}$.
- Une 4-face d'un polytope définie par un hyperplan h_{l^*} représente un ensemble de droites de dimension 3 (3-trace) incidentes à la droite l .

Un complexe de polytopes induit une partition des droites réelles en fonction des objets qu'elles intersectent. Une approche globale réalise une partition des segments en fonction des objets qu'ils interceptent. Cette différence a deux implications principales :

- En visibilité globale, les segments d'un ensemble (et par extension les droites critiques qui les supportent) sont tous incidents à leurs générateurs (arêtes/sommets). En revanche, les droites réelles représentées par une k -face d'un polytope ne sont pas nécessairement incidentes à des arêtes, mais à leurs droites supports. De fait, les bords d'un polytope ne représentent pas seulement des ensembles de droites effectivement contraintes par la géométrie de la scène dans \mathbb{R}^3 .
- De plus, prenons l'exemple d'une droite qui intersecte successivement les objets $\{A_1, \dots, A_n\}$. Le long de cette droite, une approche globale distingue tous les segments interceptant les couples d'objets (A_i, A_{i+1}) ($i < n$). Cette distinction n'est pas possible dans un espace de droites. D'une certaine façon, tous les objets deviennent transparents. Pour cette raison, un complexe de polytopes échoue à représenter la visibilité entre plusieurs couples d'objets.

Néanmoins, des rapprochements sont possibles si l'on se limite à la visibilité de deux polygones A et B . Dans ce cas, l'ensemble des segments qui les interceptent et l'ensemble des droites qui les intersectent sont équivalents. Les ensembles de segments critiques qui forment le bord de la cellule $\mathcal{C}_{A,B}$ sont contenus dans les bords du complexe de polytopes équivalent. Contenus seulement, puisque comme nous venons de le souligner, les ensembles de droites critiques sont prolongés le long des droites supports aux arêtes de la scène. Dans ce cas, nous parlerons d'ensemble critique et d'événements visuels étendus. La figure 1.19 illustre ce propos.

1.4 Précédents travaux

Cette section présente un état de l'art ciblé. L'accent est mis sur les travaux en rapport avec le calcul d'une représentation de la visibilité. En particulier, nous détaillons les travaux effectués par Nirenstein et Bittner sur lesquels nous nous appuyons par la suite.

1.4.1 Arrangement d'hyperplans

La résolution du problème présenté au paragraphe 1.2 est soumise à une complexité théorique temps et mémoire en $O(n^4 \log n)$ où n est le nombre de bloqueurs considérés.

Ce résultat est lié aux classes d'isotopie induites par un ensemble de droites. Le problème se décrit usuellement comme suit [Pel93] : on considère un ensemble de droites bleues et deux droites rouges. Ces dernières appartiennent à la même classe d'isotopie si la première peut être déplacée et confondue avec la seconde sans jamais croiser ou devenir parallèle à une droite bleue. Dans l'espace de Plücker, les hyperplans associés aux droites bleues forment un arrangement dont les cellules (de dimension 5) correspondent aux classes d'isotopie. Plus exactement, il s'agit des cellules qui intersectent la quadrique de Plücker.

D'une manière générale, un arrangement de n hyperplans dans \mathbb{R}^d induit un nombre maximum de cellules $O(n^d)$ [EOS86]. Restreint à la zone d'une surface algébrique de dimension $(d - 1)$ (*i.e.* les cellules qui intersectent cette surface), le nombre maximum de cellules devient $O(n^{d-1} \log n)$ [APS93]. D'où une complexité en $O(n^4 \log n)$ pour les classes d'isotopie induites par un ensemble de droites réelles dans l'espace de Plücker.

Si les droites considérées sont supports des arêtes d'un ensemble de polygones convexes et orientés, les droites intersectant un même sous-ensemble de ces polygones appartiennent à la même classe d'isotopie. La visibilité depuis un polygone a pour complexité $O(n^4 \log n)$ puisque le complexe de polytopes calculé est, dans le pire des cas, l'arrangement d'hyperplans associés aux arêtes des bloqueurs dans la zone de la quadrique de Plücker.

On peut noter que Pellegrini propose une complexité équivalente en $O(n^{4+\epsilon})$ [Pel90, Pel91]. Le problème des classes d'isotopie est connexe à d'autres problématiques liées aux droites dans \mathbb{R}^3 . Par exemple, étant supposé construit un arrangement d'hyperplans dans l'espace de Plücker, Pellegrini montre que l'on peut répondre au problème du lancer de rayons en $O(\log n)$ [Pel93]. Un tour d'horizon plus complet de ces problématiques peut être trouvé dans [Pel97].

1.4.2 Le graphe d'aspect

Un graphe d'aspect a pour but la représentation de toutes les vues possibles d'un objet. En ce sens, il est la représentation duale de la partition des points de vue en cellules pour lesquelles l'aspect d'un objet est qualitativement invariant. Cette approche a été introduite en vision par ordinateur pour la reconnaissance d'objets [KvD76, KvD79]. Un robot peut alors comparer la vue qu'il a d'un objet à celles représentées dans un graphe d'aspect. L'objet est identifié si l'une d'entre elles concorde.

La notion d'événement visuel provient des travaux autour des graphes d'aspect. L'aspect d'un objet change lorsque l'observateur traverse un événement visuel dû à la géométrie de l'objet. Aussi, les noeuds d'un graphe d'aspect sont les vues possibles d'un objet, ses arcs représentent les événements visuels qui délimitent deux cellules de vues.

Le problème qu'un graphe d'aspect se veut résoudre est complexe. L'aspect d'un objet ne se réduit pas à

sa silhouette, ni à ses parties visibles. Pour des polyèdres quelconques, la complexité d'un graphe d'aspect atteint $O(n^6)$ en projection orthographique et $O(n^9)$ avec une projection perspective.

Sur un plan pratique, la construction d'un graphe d'aspect est généralement approximative. L'espace des points de vue est échantillonné, puis les vues similaires regroupées pour constituer les noeuds du graphe d'aspect [HK85]. Cette solution pose néanmoins le problème du nombre d'échantillons à considérer ainsi que la possibilité d'obtenir un ensemble de vues incomplet. Si la construction exacte d'un graphe d'aspect fait l'objet de recherches [GM90, GCS91], elle demeure encore sans solution pratique.

Un tour d'horizon plus complet sur les graphes d'aspect est disponible dans [Dur99]¹, et un état de l'art sur le sujet est consultable dans [EBD92].

1.4.3 L'Asp

L'asp [Pla88] (pour *aspect representation*) a pour but de représenter l'apparence d'un objet en fonction des points d'observation possibles. L'asp d'un polygone est l'ensemble des droites qui l'intersectent. Cette définition permet de traiter les problèmes d'occultation en raisonnant sur des ensembles [PD90]. Ainsi l'occultation engendrée par plusieurs polygones est l'union de leurs asp. De même l'occultation d'un polygone P_1 vis à vis d'un polygone P_2 se traduit par la soustraction de leurs asp.

L'espace dans lequel l'asp est défini dépend de la projection considérée. Pour une projection orthographique, il s'agit de l'espace 4D des droites. L'espace 5D des rayons sera utilisé en projection perspective.

L'asp est appliqué à la construction de graphes d'aspect pour des polyèdres [PD86]. D'autres travaux concernent la reconnaissance d'objet, ou encore la maintenance de vue pour des objets polyédriques en rotation autour d'un axe [PDS90].

Cependant, l'utilisation pratique de l'asp se heurte à l'absence d'implantation robuste (à ce jour). Néanmoins, en ramenant le problème de l'occultation à la soustraction d'ensembles de droites, l'asp posait déjà un principe repris par les approches basées sur la paramétrisation de Plücker.

1.4.4 Portails et antipénombre

Teller [Tel92a] pose le problème de la visibilité à travers une séquence de polygones convexes, nommés *portails*. Son approche est conforme aux principes présentés en 1.3.1. Chaque arête de polygones est orientée de manière consistante et plongée dans l'espace de Plücker sous la forme d'un hyperplan. L'ensemble des hyperplans forme un polytope dans \mathbb{P}^5 dont l'intersection avec la quadrique de Plücker est l'ensemble des droites intersectant tous les polygones de la séquence.

Utilisant cette caractérisation, Teller *et al.* [TH91] propose un algorithme pour déterminer en $O(n^2)$ si un tel ensemble existe (n est le nombre total d'arêtes), *i.e.* si le polytope associé existe. Dans ce dernier cas, des représentants de l'ensemble sont calculés en intersectant [TH99] les 1-faces du polytope avec la quadrique de Plücker. Autrement dit, il s'agit des *esl* générées par la séquence de polygones. La figure 1.20 montre une séquence de 5 polygones convexes ainsi que l'ensemble des *esl* générées.

Considérant l'un des polygones extrémités de la séquence comme une source lumineuse, Teller [Tel92b] étend l'approche au calcul de l'antipénombre à travers une séquence d'ouvertures. La figure 1.21 donne une illustration 2D de l'antipénombre. Dans ce contexte, les intersections des 2-faces avec la quadrique de Plücker sont examinées afin d'identifier les événements VE et EEE qui délimitent l'antipénombre. Une paramétrisation des surfaces associables est utilisée pour leur représentation dans \mathbb{R}^3 .

¹<http://people.csail.mit.edu/fredo/THESE/>

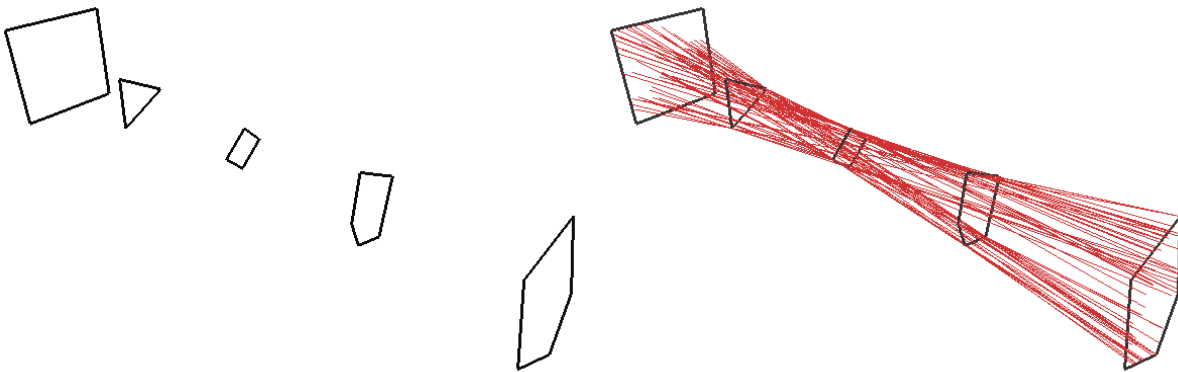


FIG. 1.20 – Visibilité à travers une séquence de polygones convexes. A gauche, la séquence en question, à droite, les *esl* de la séquence. Chacune de ces *esl* est le produit de l'intersection des 1-faces du polytope associables à la séquence de polygones dans l'espace de Plücker (captures d'écran réalisées sur la base d'une implémentation équivalente à [TH91])

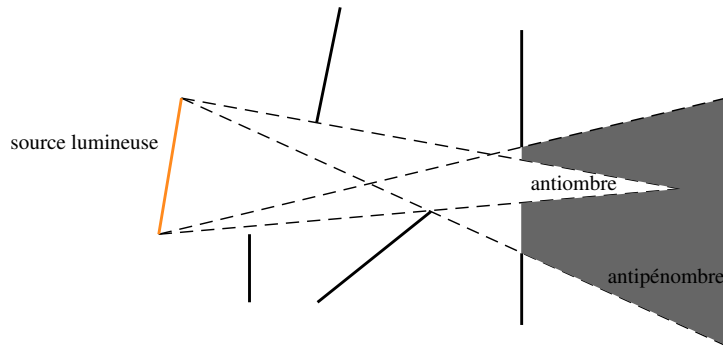


FIG. 1.21 – Illustration 2D de l'antipénombre. Le segment de gauche est assimilé à une source lumineuse (schéma adapté de [Tel92b])

Dans une scène *indoor*, le calcul de l'antipénombre est utilisé pour tester la visibilité des pièces situées aux extrémités d'une séquence d'ouvertures. Tout ou partie des pièces contenues dans l'antipénombre est visible. Le procédé permet alors d'accélérer efficacement un calcul d'illumination globale. Dans ce contexte, une approche simplifiée est présentée dans [TH93], qui ne nécessite que des comparaisons points/hyperplans dans l'espace de Plücker.

1.4.5 Le complexe de visibilité 2D

Pocchiola et Vegter définissent le complexe de visibilité 2D [PV93]. Cette structure constitue une description globale de la visibilité dans le plan. Il s'agit d'un complexe cellulaire qui partitionne les segments du plan en fonction des objets qu'ils interceptent à leurs extrémités. Des algorithmes de construction du complexe de visibilité 2D ont été proposés pour des objets convexes lisses [PV96] ou polygonaux [Riv95].

Le complexe de visibilité 2D a fait l'objet d'applications en simulation d'éclairage dans des environne-

ments dynamiques [DORP96, ORDP96]. Il a également été utilisé pour la maintenance de vue [Riv97a, Riv97b]. Il est la principale source d'inspiration pour l'étude globale de la visibilité en 3D.

1.4.6 Le complexe de visibilité 3D

Durand *et al.* [Dur99] réalise hors contexte applicatif une étude globale de la visibilité tridimensionnelle. S'inspirant des travaux de Pocchiola et Vegter [PV93], Durand décrit le complexe de visibilité 3D [DDP02], adaptation tridimensionnelle du complexe de visibilité 2D. Cette structure constitue une partition des segments qui interceptent les mêmes objets à leurs extrémités.

L'idée générale est de regrouper en composantes connexes les droites qui intersectent les mêmes objets. Les ensembles ainsi constitués sont partitionnés en ensembles de segments dont les extrémités interceptent le même couple d'objets. On parle aussi de segments libres maximaux, au sens où un tel segment n'intersecte aucun autre objet en son intérieur, et est de taille maximale. Cette construction s'appuie sur une paramétrisation 4D des droites de \mathbb{R}^3 , et une pseudo dimension additionnelle utilisée pour distinguer les ensembles de segments.

Le complexe de visibilité 3D est constitué de faces de dimension 0 à 4. La dimension d'une face est le nombre de degrés de liberté que possèdent les droites/segments qu'elle représente. Par exemple les 0-faces correspondent aux *esl* de la scène. Durand indique que la taille du complexe de visibilité 3D est $O(n^4)$ où n est le nombre d'objets dans la scène. Sur un plan applicatif, l'utilisation du complexe de visibilité reste problématique. L'algorithme de construction proposé s'avère manquer de robustesse face aux cas de dégénérescences géométriques, *i.e.* les cas d'alignements de sommets ou encore de faces coplanaires.

Goaoc [Goa04] poursuit l'étude du complexe de visibilité 3D, qu'il redéfinit afin de prendre en compte les cas de dégénérescences. Un nouvel algorithme de construction du complexe est proposé pour un ensemble d'objets courbes. En outre, Goaoc montre que la taille du complexe de visibilité n'est pas nécessairement dissuasive, et établit que celle-ci augmente en moyenne linéairement avec le nombre d'objets s'ils sont uniformément distribués dans la scène. Pour k polytopes (dans \mathbb{R}^3) avec un total de n arêtes, il montre que la taille du complexe est au pire $O(n^2 k^2)$ [BDD⁺02, BDD⁺04]. Sur le plan pratique, aucune application n'est rapportée.

1.4.7 Le squelette de visibilité

Une description partielle de la visibilité s'avère suffisante pour certaines applications. C'est le cas par exemple pour le calcul d'un maillage de discontinuités où seules les limites d'ombres et de pénombres sont recherchées. Le squelette de visibilité [DDP97] est une représentation partielle du complexe de visibilité 3D, d'où le terme *squelette*. Seules les faces de dimensions 0 et 1 y sont encodées, *i.e.* les *esl* et les surfaces d'événements visuels de type VE ou EEE. Concrètement, le squelette de visibilité est un graphe où chaque noeud est associé à une *esl* de la scène, et chaque arc à un événement visuel de type VE ou EEE. Un arc relie deux noeuds si les deux *esl* associées délimitent l'événement porté par l'arc. La figure 1.22 illustre cette structure.

Durand propose un algorithme de construction du squelette de visibilité basé sur un catalogue de configurations géométriques pouvant engendrer une *esl* ou un événement VE ou EEE. Le squelette de visibilité est alors construit en parcourant la géométrie de la scène à la recherche des configurations présentes dans ce catalogue. Plus robuste que le complexe de visibilité 3D, le squelette de visibilité est utilisé pour des calculs d'illumination globale [DDP98, DDP99].

Cependant, parce qu'il se base sur un catalogue de configurations types, l'algorithme proposé par Durand supporte mal les cas de dégénérescences géométriques qui démultiplient à l'infini les combinaisons générant un événement visuel. La figure 1.23 donne un exemple d'un événement VE dégénéré.

Pour pallier ce problème, Duguet *et al.* [DD02, Dug04], propose le squelette de visibilité robuste. Leur algorithme de construction supprime l'utilisation d'un catalogue, et gère les dégénérescences géométriques. La contrepartie est la détection des noeuds et arcs du squelette à un epsilon près.

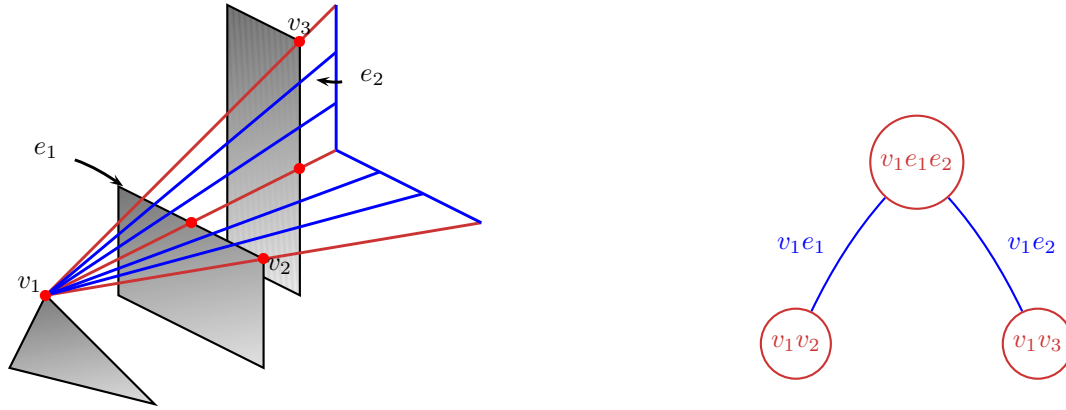


FIG. 1.22 – Le squelette de visibilité encode les relations d'incidence entre les *esl* et les événements de type VE ou EEE. Ici, l'événement v_1e_1 (resp. v_1e_2) est incident aux événements $v_1e_1e_2$ et v_1v_2 (resp. v_1v_3).

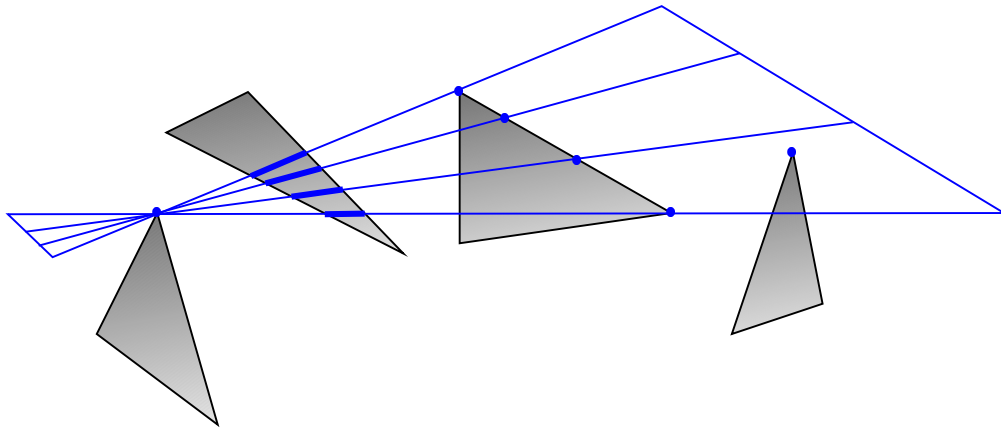


FIG. 1.23 – Un exemple d'événement VE dégénéré. Plus d'un sommet et d'une arête définissent le même événement.

1.4.8 Arbre PSP

Pour représenter l'ensemble des visibilitées depuis un polygone, Pu *et al.* [Pu98, MP99] proposent la construction d'un arbre PSP (*Plücker Space Partition*), i.e. un arbre BSP (*Binary Space Partition*) dans

l'espace de Plücker. Les feuilles d'un arbre PSP représentent un ensemble de droites bloquées ou non.

Initialement, la racine d'un arbre PSP représente l'ensemble des droites issues d'un polygone source. L'algorithme de construction proposé consiste à insérer successivement dans l'arbre les hyperplans associés aux arêtes d'un bloqueur. Pour chaque nœud interne rencontré, l'algorithme teste si l'hyperplan intersecte la région de l'espace qu'il représente. Si ce test réussit, il est répété dans les fils. Lorsqu'une feuille est atteinte, c'est que l'hyperplan subdivise la région représentée. La feuille est remplacée par un nœud interne contenant l'hyperplan inséré, et possédant deux nouvelles feuilles produits de la subdivision.

L'inconvénient principal de cet algorithme est que les intersections sont locales à la zone des hyperplans insérés et non à la zone du bloqueur. Cela revient à calculer l'arrangement des hyperplans associés aux bloqueurs et à isoler les cellules qui correspondent aux droites non bloquées. La construction d'un arbre PSP est donc directement soumise à la complexité théorique $O(n^5)$ d'un arrangement d'hyperplans dans l'espace de Plücker.

Les résultats des expériences confirment ce comportement. Pu *et al.* déterminent une complexité pratique en $O(n^{5.17})$. Après avoir supprimé toutes les régions de l'arbre PSP qui ne possèdent pas d'intersection avec la quadrique de Plücker, cette complexité est ramenée en $O(n^{4.31})$. Ce dernier résultat est conforme à la complexité $O(n^4 \log n)$ d'un arrangement d'hyperplans au voisinage de la quadrique de Plücker. L'algorithme, testé sur des scènes n'excédant pas 15 triangles, s'avère donc peu praticable. Toutefois, ces travaux sont précurseurs à ceux de Nirenstein et plus particulièrement de Bittner.

1.4.9 Cellule de vue exacte

Les travaux réalisés par Nirenstein [NBG02] constituent la première solution pratique au problème de la visibilité exacte entre polygones. Bien que le but de son approche soit l'existence d'au moins une visibilité, celle-ci procure également le moyen de calculer une information de visibilité globale entre polygones. Comme indiqué précédemment, il n'existe pas à ce jour d'algorithme pour prouver exactement la visibilité de 2 polygones sans devoir montrer, le cas échéant, que l'ensemble de toutes les visibilités est vide ou non vide.

Nirenstein met en place les principes présentés en 1.3.2. L'ensemble des droites intersectant deux polygones est caractérisé par un polytope dans l'espace de Plücker. Les droites occultées sont ensuite supprimées par soustraction des polyèdres associés à chaque bloqueur.

Un algorithme pour le calcul de la V-représentation d'un polytope associable à deux polygones peut être trouvé dans [Nir03]. Considérant le polyèdre associable à deux polygones convexes A et B , Nirenstein montre qu'un hyperplan de projection peut être choisi tel que ses arêtes soient toutes parallèles deux à deux, comme l'illustre la figure 1.24. Notant \vec{r} un vecteur colinéaire aux arêtes, le polyèdre est donc ouvert dans les directions \vec{r} et $-\vec{r}$. L'intersection de chacune de ses arêtes avec la quadrique de Plücker est une *esl* générée par un sommet de A et un sommet de B . Pour transformer le polyèdre en polytope, deux hyperplans H_{\perp}^+ et H_{\perp}^- orthogonaux au vecteur \vec{r} sont ajoutés de part et d'autre de la quadrique de Plücker comme illustré sur la figure 1.25.a. Les sommets du polytope ainsi obtenu sont calculables en $O(nm)$, où n et m sont les nombres de sommets respectifs des polygones A et B . Les droites sommet à sommet sont énumérées et leurs points de Plücker sont projetés orthogonalement sur les hyperplans H_{\perp}^+ et H_{\perp}^- . La figure 1.25.b illustre cette construction.

L'algorithme précédent est démontré dans le cas où l'un des deux polygones est contenu dans le plan $x = 0$ (l'hyperplan de projection adéquat dans l'espace de Plücker est alors $\pi_0 = 0$). Puisqu'il est

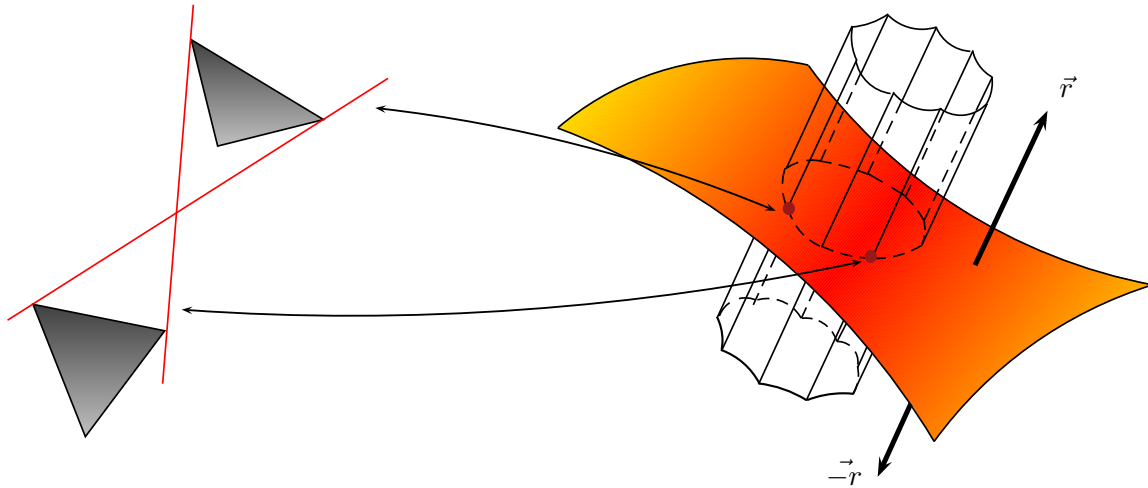


FIG. 1.24 – Considérant deux polygones (convexes), un hyperplan de projection peut être choisi tel que les arêtes du polyèdre associé aient la même direction \vec{r} . Le polyèdre est ouvert dans les directions \vec{r} et $-\vec{r}$. L'intersection de chaque arête avec la quadrique est une droite sommet à sommet des deux polygones. L'exemple ci-dessus avec deux triangles présente donc 9 arêtes.

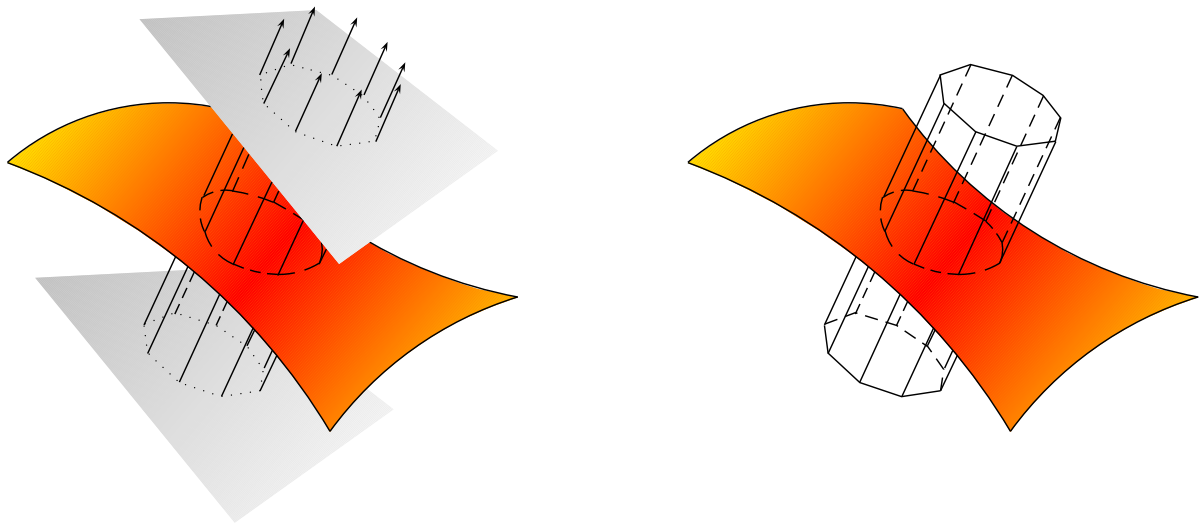


FIG. 1.25 – A gauche : pour obtenir un polytope, deux hyperplans H_{\perp}^{+} et H_{\perp}^{-} orthogonaux à \vec{r} sont ajoutés de part et d'autre de la quadrique. A droite : les sommets du polytope résultant sont les projetés sur H_{\perp}^{+} et H_{\perp}^{-} des droites sommet à sommet des deux polygones.

toujours possible de se ramener à cette configuration par transformation géométrique, le principe de l'algorithme demeure valide en général. Cependant, \vec{r} , H_{\perp}^{+} et H_{\perp}^{-} ne sont déterminés que pour le cas $x = 0$. Cette limitation est levée par Haumont [HMN05] (voir 1.4.11).

Pour effectuer les opérations CSG dans l'espace de Plücker, Nirenstein s'appuie sur l'algorithme de Bajaj et Pascucci [BP96] permettant d'intersecter un complexe de polytopes par un hyperplan. Cet al-

gorithme est indépendant de la dimension de l'espace utilisé. Il présente en outre l'avantage de limiter les opérations numériques au strict minimum. En revanche il nécessite de calculer au préalable le graphe d'incidence du polytope.

Nirenstein applique ses travaux au calcul exact de PVS (*Potentially Visible Set*) [ARFPB90], qui consiste à déterminer exactement les polygones visibles depuis toutes les faces d'une cellule de vue. La visibilité d'un polygone avec une face d'une cellule est établie en calculant l'ensemble de toutes leurs visibilités dans l'espace de Plücker, afin de montrer que celui-ci est vide ou non-vide. Dans ce dernier cas, le complexe de polytopes associé n'est pas conservé, puisque seule la preuve de la visibilité est recherchée. Parce qu'elles sont consommatrices de temps, Nirenstein développe une architecture logicielle pour limiter le recours aux opérations CSG dans l'espace de Plücker. Une subdivision spatiale et hiérarchique de la scène est utilisée pour hiérarchiser les requêtes de visibilité depuis une cellule de vue. L'intérêt est de ne pas tester individuellement les polygones contenus dans les régions détectées invisibles. De plus, les faces de ces dernières sont utilisées comme bloqueurs virtuels. L'idée est de réduire le nombre de bloqueurs en les remplaçant par les faces de la région qui les contient. Enfin une première requête agressive de visibilité est utilisée en lançant quelques rayons entre les deux polygones. En cas d'échec, les bloqueurs sont ordonnés en fonction du nombre de rayons qui les intersectent. Ce mécanisme vise à soustraire en priorité les bloqueurs dont le pouvoir occultant est le plus grand.

1.4.10 Arbre d'occultation

Comme Nirenstein, l'approche suivie par Bittner repose sur les principes présentés en 1.3.2. Il développe une structure baptisée *arbre d'occultation*. Comme les arbres PSP (voir 1.4.8), il s'agit d'un arbre BSP dans l'espace de Plücker. Les deux structures diffèrent cependant sur deux points. D'une part, un arbre d'occultation comprend une information de profondeur et, d'autre part, sa construction s'éloigne du calcul d'un arrangement d'hyperplans.

Bittner propose initialement les arbres d'occultation dans \mathbb{R}^3 pour représenter la visibilité depuis un point de vue en 3D [BHS98]. Le principe est de traiter les polygones du plus proche au plus lointain. Pour chaque polygone, son volume d'ombre (polyèdre convexe délimitant la région de l'espace masquée par le polygone depuis le point d'observation) est calculé et inséré [Nay90, NAT90] dans l'arbre d'occultation. Les fragments de polygones atteignant une feuille représentant une région invisible de l'espace sont supprimés. Si c'est une région visible qui est atteinte, la feuille est remplacée par la représentation BSP du volume d'ombre dû au fragment. La figure 1.26 illustre cette structure. Elle peut aussi être considérée comme une représentation de toutes les droites issues du point de vue et regroupées en fonction du premier polygone intersecté (d'où l'information de profondeur). Par la suite, Bittner étend les arbres d'occultation pour représenter la visibilité depuis une région en 2D et 2.5D [BP01, BWW05], puis depuis un polygone en 3D [Bit02].

Dans le cas 3D, l'arbre BSP est construit dans l'espace de Plücker. La représentation BSP rend immédiate la généralisation de l'algorithme. L'arbre est initialisé avec une feuille représentant toutes les droites issues du polygone source. Puis, les bloqueurs sont traités par ordre croissant d'éloignement. Pour chacun d'entre eux, le polytope associé au polygone source et au bloqueur est calculé. Celui-ci est ensuite inséré dans l'arbre BSP, pour y être successivement intersecté par les hyperplans contenus dans les noeuds rencontrés. Lorsqu'un fragment du polytope atteint une feuille marquée visible, celle-ci est remplacée par la représentation BSP de ce fragment. Si la feuille est marquée invisible (*i.e.* elle représente un ensemble de droites déjà bloquées), le fragment est supprimé. Lorsqu'il ne reste plus de bloqueurs à traiter, l'arbre d'occultation représente les droites issues du polygone source et regroupées dans l'espace de Plücker en ensembles convexes et en fonction du premier bloqueur intersecté.

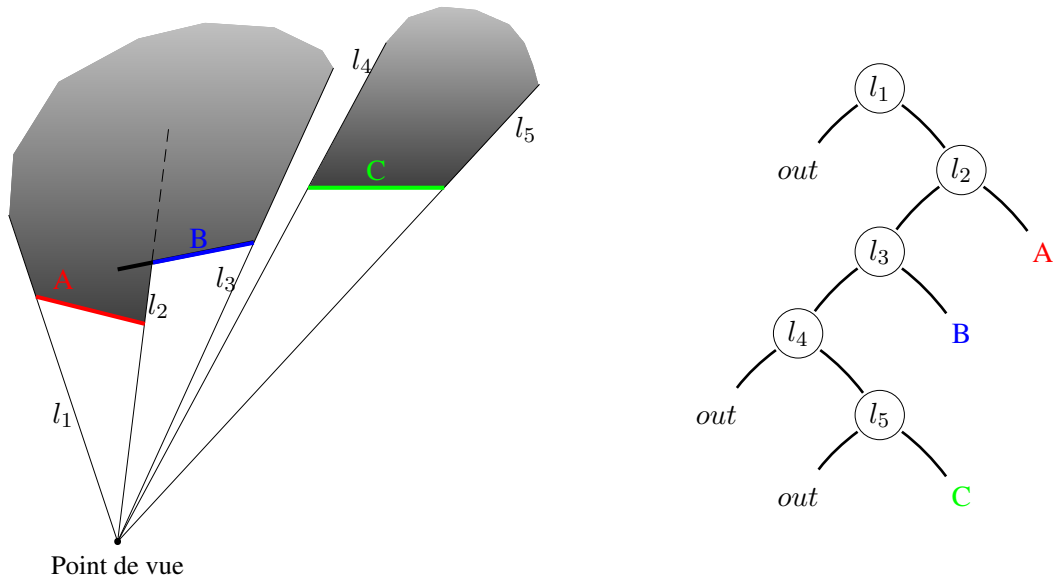


FIG. 1.26 – Illustration 2D d'un arbre d'occultation pour un point de vue. Les bloqueurs sont insérés dans l'arbre BSP par ordre d'éloignement croissant en étant successivement intersectés par les hyperplans contenus dans les nœuds rencontrés. Lorsqu'une feuille visible est atteinte, elle est remplacée par la représentation BSP de l'ombre engendrée par le bloqueur ou par le fragment de bloqueur depuis le point d'observation. Si la feuille représente un ensemble de droites déjà bloquées, le fragment est supprimé. Ainsi sur ce schéma, les bloqueurs A et C et le fragment B sont déterminés visibles. (Illustration adaptée de [BHS98])

Pour effectuer l'intersection d'un polytope par un hyperplan, Bittner n'utilise pas les mêmes algorithmes que Nirenstein. Son implantation repose uniquement sur un algorithme d'énumération similaire à [AF96]. À proprement parler, aucun calcul explicite d'intersection d'un polytope par un hyperplan n'est effectué. Si une intersection existe, l'hyperplan est ajouté (deux fois en inversant son orientation) à la H-représentation du polytope, puis tous les sommets sont à nouveau énumérés. Les V-représentations des deux polytopes produits de l'intersection sont ainsi obtenues. Si cette solution est plus simple à mettre en œuvre que celle retenue par Nirenstein, elle est aussi plus coûteuse en terme de temps de calcul. En effet, elle recalcule tous les sommets et pas seulement ceux nouvellement créés par l'intersection.

Un arbre d'occultation forme une représentation hiérarchique de la visibilité depuis un polygone source. Tout comme un arbre PSP, sa complexité au pire est $O(n^4 \log n)$ (complexité d'un arrangement d'hyperplans au voisinage de la quadrique de Plücker). Cependant, dans le cas d'un arbre PSP, la traversée d'un nœud consiste à tester l'intersection de la région qu'il représente avec un hyperplan de la H-représentation d'un bloqueur. Pour un arbre d'occultation, c'est l'intersection du polytope d'un bloqueur avec l'hyperplan contenu dans un nœud qui est testée. Cette différence fait que les intersections se font dans la zone des hyperplans rencontrés dans les nœuds, alors que pour un arbre PSP elles se font dans la zone des hyperplans insérés. En cela, la construction s'éloigne du coût prohibitif d'un arrangement d'hyperplans.

Comme Nirenstein, Bittner applique ses travaux au calcul exact de PVS. Un arbre d'occultation est calculé pour chaque face d'une cellule de vue afin de déterminer l'ensemble des polygones visibles. Une subdivision spatiale et hiérarchique (ici un *kd-tree*) de la scène est également utilisée afin de déterminer les régions de l'espace invisibles.

1.4.11 Preuve de visibilité

Récemment, Haumont *et al.* [HMN05] ont publié des travaux qui améliorent significativement, et ce sur plusieurs points, ceux de Nirenstein.

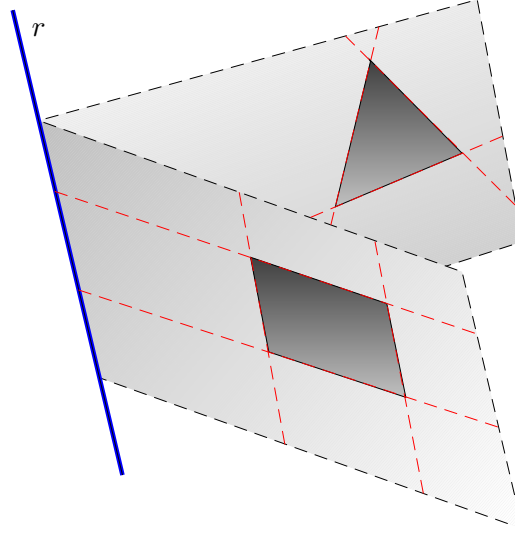


FIG. 1.27 – La droite r est l'intersection des plans supports aux deux polygones. Toutes les droites supports des arêtes des deux polygones intersectent ou sont parallèles à r . Les hyperplans associés sont donc tous incidents à r^* dans l'espace de Plücker. Le cas où les plans supports des polygones sont eux-mêmes parallèles correspond à une intersection à l'infini. Dans ce cas r^* vaut $(0, 0, 0, -n_x, -n_y, -n_z)$ avec (n_x, n_y, n_z) les coefficients de la normale des deux plans (illustration adaptée de [HMN05]).

Tout d'abord, l'algorithme d'énumération en $O(nm)$ proposé par Nirenstein (voir 1.4.9) est généralisé à toute configuration. Il devient donc utilisable quelle que soit la position des deux polygones dans l'espace, et ne nécessite plus de transformations géométriques préalables. Haumont considère la droite r à l'intersection des plans supports aux deux polygones. Elle est incidente (ou parallèle) aux droites e_i supports des arêtes des polygones. La figure 1.27 en donne une illustration. Dans l'espace de Plücker, les hyperplans $h_{e_i}^*$ sont donc tous incidents au point de Plücker r^* . Considérant un hyperplan de projection quelconque, le polyèdre de \mathbb{R}^5 obtenu est une pyramide infinie de sommet r_p^* puisque tous les hyperplans s'intersectent en ce point. Pour que ce polyèdre ait toutes ses arêtes parallèles deux à deux, Haumont en déduit que l'hyperplan de projection doit être choisi tel que r_p^* soit projeté à l'infini. Dès lors, tout hyperplan parallèle au vecteur \vec{r}^* dans \mathbb{P}^5 convient. La suite de l'algorithme est similaire à Nirenstein. Deux hyperplans orthogonaux à \vec{r}^* sont ajoutés de part et d'autre de la quadrique de Plücker ; la V-représentation est finalement obtenue par projection orthogonale des *esl* sur ces hyperplans.

Concernant les intersections polyèdres/hyperplans, Haumont *et al.* proposent une restriction de l'algorithme de Bajaj et Pascucci [BP96] au 1-squelette du polytope. L'avantage de cette solution est de ne pas devoir calculer le graphe d'incidence complet du polytope associé à deux polygones. Le coût de cette étape est en effet non négligeable (nous reviendrons sur ce point dans le prochain chapitre). En contrepartie, l'information topologique partielle ne permet pas d'identifier les arêtes nouvellement créées. Un calcul supplémentaire est introduit pour les identifier en se basant sur la description combinatoire des sommets.

La silhouette d'un objet depuis un point de vue est l'ensemble des arêtes qui forment visuellement le

bord de l'objet, *i.e.* définissant un événement VE. La silhouette d'un objet est moins complexe que son maillage de polygones [McG04]. Cette notion se généralise à la vue depuis un polygone. Haumont *et al.* l'utilisent pour réduire le nombre d'opérations CSG dans l'espace de Plücker. Considérons le calcul des visibilité de deux polygones A et B partiellement masqués par un objet. Au lieu de soustraire un à un chaque polygone constitutif de l'objet, le graphe d'adjacence de son maillage est tout d'abord parcouru pour extraire les arêtes silhouettes pour A et pour B . Seuls les hyperplans associés à ces arêtes silhouettes seront utilisés pour intersecter le complexe de polytopes dans l'espace de Plücker.

Ces optimisations sont incluses au sein d'une architecture logicielle dédiée au calcul exact de PVS. Elle diffère principalement de celle de Nirenstein sur l'utilisation d'une requête agressive de visibilité. Haumont introduit une sélection des bloqueurs guidée par la visibilité. Pour chaque polytope, une droite représentative est choisie, *i.e.* un point sur la quadrique de Plücker contenu dans le polytope. Par propriété de convexité, une combinaison linéaire des sommets du polytope donne un point intérieur. Choisisant ainsi deux points p_1 et p_2 de part et d'autre de la quadrique, l'intersection de la droite (p_1p_2) avec la quadrique procure une droite représentative. Cette droite donne la direction d'un rayon lancé depuis l'un des deux polygones. Si le second polygone est atteint, c'est une visibilité et la requête prend fin. Sinon, le bloqueur à soustraire est choisi parmi l'ensemble des bloqueurs intersectés (selon une heuristique similaire à celle utilisée par Nirenstein, sélectionnant le bloqueur dont le pouvoir occultant est le plus grand). Cette technique possède deux avantages. D'une part, elle garantit que le bloqueur choisi possède bien une intersection avec le polytope courant. D'autre part, elle permet de lancer des rayons qui assurément évitent les bloqueurs qui ont déjà été soustraits. Ainsi les chances de lancer une droite représentative dans un espace de visibilité augmentent progressivement.

Cette architecture permet à Haumont de traiter des scènes de l'ordre de 4.6 millions de triangles. Les mesures réalisées montrent que la complexité pratique pour prouver la visibilité de deux polygones est en $O(n^{1.44})$, où n est le nombre de bloqueurs effectivement traités.

1.5 Conclusion

Les travaux en visibilité globale ont démontré l'importance d'accroître notre compréhension de la visibilité ainsi que l'utilité de disposer d'une information plus complète que la seule preuve de la visibilité de primitives géométriques. Le complexe de visibilité 3D (voir 1.4.6) est une structure qui va en ce sens. Son étude a amélioré la compréhension du phénomène visuel et permis de créer de nouvelles solutions pratiques telles que le squelette de visibilité (voir 1.4.7). Cependant, ce dernier reste une représentation partielle de la visibilité.

Si les constructions opérant dans l'espace de Plücker ne sont pas des structures de visibilité globale (voir 1.3.3), elles permettent cependant une description de toutes les visibilitées entre deux objets. Cette information dépasse la simple preuve.

D'abord utilisée d'un point de vue purement formel par Pellegrini (voir 1.4.1), la paramétrisation de Plücker démontre son efficacité pratique dans le cadre des travaux de Teller (voir 1.4.4), relatifs au calcul de l'antipénombre. Les solutions au problème de la visibilité de polygones à travers un ensemble de bloqueurs apparaissent d'abord avec les travaux de Pu (voir 1.4.8), puis surtout avec ceux de Nirenstein (voir 1.4.9), et de Bittner (1.4.10).

Tous ces travaux reposent sur les mêmes fondamentaux : ils calculent tous une structure équivalente à un complexe de polytopes dans l'espace de Plücker. En dépit de ces points communs, de nombreuses différences apparaissent, tant au niveau des structures construites qu'au niveau des solutions algorithmiques utilisées. De plus, les applications proposées n'exploitent pas totalement le potentiel de l'information calculée. Dans tous les cas, les précédents travaux ne retiennent que la preuve exacte de l'existence ou non d'une visibilité.

Les travaux présentés dans ce mémoire partent de ces observations. Notre premier objectif est de proposer une synthèse des approches existantes, et, le cas échéant, de pallier leurs faiblesses, afin de proposer une solution adéquate au calcul d'une représentation complète des visibilitées de deux polygones convexes. Ces travaux font l'objet du chapitre 2. Notre second objectif est de proposer des solutions afin d'exploiter pleinement l'information calculée, et de démontrer son intérêt, aussi bien pour le calcul d'ombres douces, qu'en radiocommunications. Ces travaux font l'objet du chapitre 3 et du chapitre 4.

Visibilité polygone à polygone

Sommaire

| | | |
|------------|-----------------------------------|-----------|
| 2.1 | Problématique | 40 |
| 2.1.1 | Objectifs | 40 |
| 2.1.2 | Effet de fragmentation | 40 |
| 2.2 | Approche proposée | 45 |
| 2.2.1 | Test d'intersection renforcé | 45 |
| 2.2.2 | Arbre VBSP | 47 |
| 2.2.3 | Valider et rentabiliser | 50 |
| 2.3 | Implantation et évaluation | 52 |
| 2.3.1 | Éléments d'implantation | 52 |
| 2.3.2 | Évaluation de la méthode | 53 |
| 2.3.3 | Discussion | 56 |
| 2.4 | Conclusion | 59 |

Ce chapitre est consacré au calcul d'une représentation de la visibilité entre deux polygones convexes de \mathbb{R}^3 . La partie 2.1 étudie les problèmes induits par le calcul d'opérations CSG au voisinage de la quadrique de Plücker. Elle rappelle pourquoi certaines opérations ne sont pas utiles, et comment celles-ci engendrent un effet de fragmentation de la visibilité. Des opérations inutiles signifient des temps de calculs plus importants, et le cas échéant, une augmentation des risques d'instabilités numériques. Une fragmentation de l'information de visibilité implique une consommation accrue des ressources mémoires. La partie 2.2 présente un test d'intersection polytope/polyèdre renforcé afin de limiter ces problèmes. Elle décrit ensuite la structure d'arbre VBSP utilisée pour représenter et mémoriser la visibilité des polygones, et présente son algorithme de calcul. Celui-ci s'appuie sur le test d'intersection renforcé couplé à une synthèse des travaux de Nirenstein [NBG02] et de Bittner [Bit02]. Enfin la partie 2.3 évalue la complexité pratique de notre approche, rend compte et discute des résultats obtenus.

Une version préliminaire des travaux présentés dans ce chapitre peut être trouvée dans [MAM05].

2.1 Problématique

Cette partie détaille tout d’abord les objectifs de ce chapitre. Puis elle examine les problèmes induits par les opérations CSG au voisinage de la surface de Plücker.

2.1.1 Objectifs

Notre objectif premier est le calcul dans l’espace de Plücker d’une représentation cohérente de la visibilité entre deux polygones. C’est dans ce contexte que le terme “cohérence” est à préciser. Par cohérence, nous entendons une information qui tend vers un complexe de polytopes minimal, représentatif de l’ensemble des visibilité. En d’autres termes, il s’agit de représenter la visibilité de deux polygones à l’aide d’un minimum de polytopes. Nous souhaitons ainsi réduire la complexité pratique du calcul de visibilité et la confronter à sa borne théorique temps et mémoire $O(n^4 \log n)$ ¹. De plus, dans l’optique d’exploiter l’information calculée (chapitres 3 et 4), celle-ci devra être stockée. Diminuer la taille utile du complexe de polytopes, c’est nécessairement réduire les besoins en mémoire pour le stocker. Dans ce but, nous nous appuyons principalement sur les approches proposées par Nirenstein [NBG02] et Bittner [Bit02].

Il existe peu d’indications sur la complexité pratique de ce type d’approche. Les complexités évaluées par Nirenstein et Haumont sont celles de leurs architectures logicielles, conçues pour établir une preuve de la visibilité le plus rapidement possible. Ces architectures prouvent la visibilité en évitant autant que possible le recours aux calculs CSG dans l’espace de Plücker. Ce n’est pas la complexité du calcul complet d’un complexe de polytopes qui est donc évaluée.

Les arbres d’occultation proposés par Bittner sont des représentations de l’ensemble des visibilité. Malheureusement, les tests réalisés par Bittner [Bit02] ne sont pas suivis d’une analyse de la complexité pratique. En outre, un arbre d’occultation encode la visibilité *depuis* un polygone et comprend une information de profondeur : les ensembles de droites bloquées sont distingués en fonction du premier bloqueur qu’elles intersectent. Dans notre cas, nous nous intéressons seulement aux visibilité de deux polygones et ne faisons aucune distinction entre les différents ensembles de droites bloquées. L’information calculée n’est évidemment pas la même, et n’est donc pas directement comparable.

2.1.2 Effet de fragmentation

Cette partie considère les problèmes induits par le calcul d’opérations CSG au voisinage de la quadrique de Plücker. Il s’agit d’intersections inutilement calculées, et engendrant ce que nous appellerons la *fragmentation de la visibilité*. La conséquence est un surcoût en temps et en mémoire. Nous expliquons pourquoi l’algorithme de Nirenstein, et *a fortiori*, celui de Bittner, sont sujets à ces problèmes. De cette étude, nous concluons sur la nécessité d’améliorer l’efficacité des tests d’intersection.

Dans son approche, Nirenstein soustrait à un polytope les droites poignardantes de chaque bloqueur O . Les arêtes d’un bloqueur O induisent dans l’espace de Plücker la H-représentation du polyèdre \mathcal{P}_O soustrait au polytope. Plaçons-nous dans cette configuration :

Considérons un polytope \mathcal{P}_{AB} associé à deux polygones convexes A et B , ainsi qu’un polyèdre \mathcal{P}_O associé à un bloqueur O . Nous dirons que la soustraction de \mathcal{P}_O à \mathcal{P}_{AB} est *efficace* si :

1. $\mathcal{P}_{AB} \cap \mathcal{P}_O \cap \mathcal{Q} \neq \emptyset$. Autrement dit, les intersections respectives de \mathcal{P}_{AB} et \mathcal{P}_O avec la quadrique de Plücker (\mathcal{Q}) ne sont pas disjointes.

¹Rappel : dans notre contexte, il s’agit de la complexité d’un arrangement d’hyperplans au voisinage de la quadrique de Plücker (voir 1.4.1)

2. Un hyperplan de la H-représentation de \mathcal{P}_O qui intersecte un polytope \mathcal{P} intersecte également $\mathcal{P} \cap \mathcal{Q}$.

Autrement dit, la pertinence des intersections dépend de la capacité de l'algorithme à localiser les opérations à la zone du bloqueur soustrait. La figure 2.1 reprend l'exemple du chapitre précédent : deux triangles A et B partiellement occultés par un bloqueur O . Pour plus de clarté, la soustraction du bloqueur est illustrée par une coupe 2D transversale. Bien sûr, ce n'est qu'une vue de la réalité. Si, sur cet exemple, la soustraction du bloqueur est efficace, cela relève du cas particulier, principalement pour deux raisons :

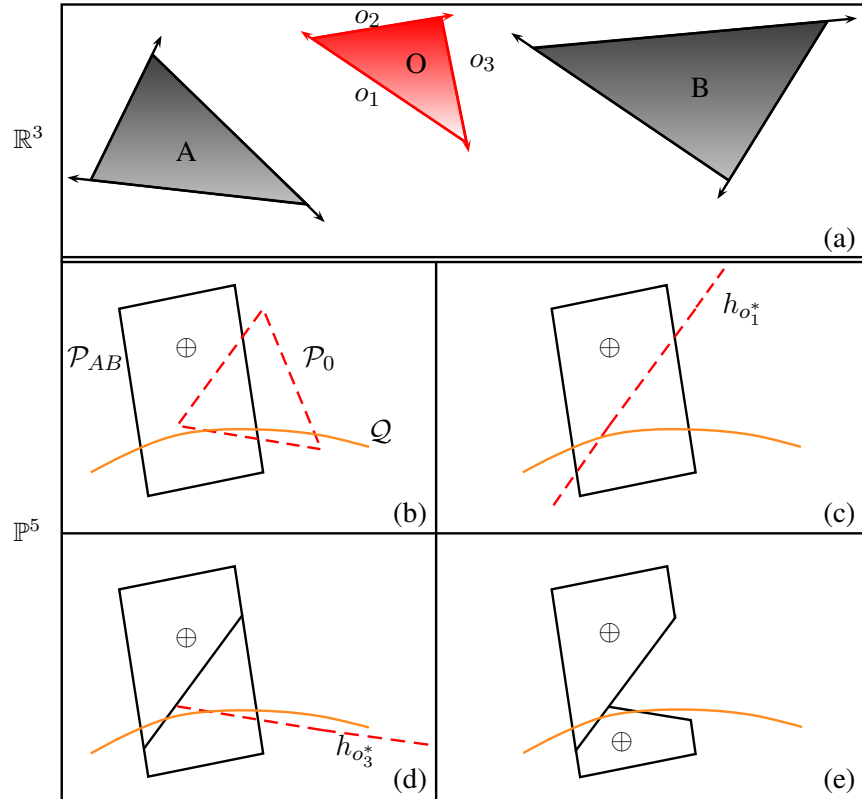


FIG. 2.1 – (a) Quelles sont les droites qui intersectent A et B mais pas le bloqueur O ? (b) Le polytope \mathcal{P}_{AB} représente l'ensemble des droites $stab(A, B)$ dans l'espace de Plücker. \mathcal{P}_O est le polyèdre associé au bloqueur O et représente $stab(O)$. Notons que $\mathcal{P}_{AB} \cap \mathcal{P}_O \cap \mathcal{Q} \neq \emptyset$. (c) \mathcal{P}_{AB} est successivement intersecté par les hyperplans de la H-représentation de \mathcal{P}_O et chaque opération affecte l'intersection du polytope avec la quadrique de Plücker (d) Le calcul se poursuit dans le demi-espace négatif de h_{o1}^* . (e) La partie commune à \mathcal{P}_O et \mathcal{P}_{AB} est supprimée. L'intersection des deux polytopes restants est l'ensemble des droites intersectant A et B mais pas O ($stab(A, B) - stab(O)$).

La première est qu'un polytope est un ensemble plus grand que les droites réelles qui nous intéressent. De part sa nature "courbe", il est difficile de travailler directement sur la quadrique de Plücker. La représentation par polytopes permet de contourner cette difficulté et de simplifier (sur le plan de l'analyse et de la mise en œuvre) les opérations CSG sur les ensembles de visibilité. En contrepartie, seule l'intersection d'un polytope avec la quadrique de Plücker constitue l'information pertinente. Ce qui rend possible les deux configurations suivantes :

- \mathcal{P}_O et \mathcal{P}_{AB} peuvent posséder une intersection non vide et néanmoins distincte de la quadrique de Plücker. La figure 2.2 illustre cette configuration. La soustraction du bloqueur est alors inutile.
- Etant supposé que $\mathcal{P}_{AB} \cap \mathcal{P}_O \cap \mathcal{Q} \neq \emptyset$, cela n'implique pas que tous les hyperplans de la H-représentation de \mathcal{P}_O donnent lieu à des intersections utiles. La figure 2.3 illustre cette possibilité.

De manière générale, toute opération qui affecte un polytope mais laisse inchangée son intersection avec la quadrique est inutile. De plus elle crée de l'information superflue, *i.e.* des polytopes qui ne représentent pas des droites réelles, et qui doivent alors être supprimés.

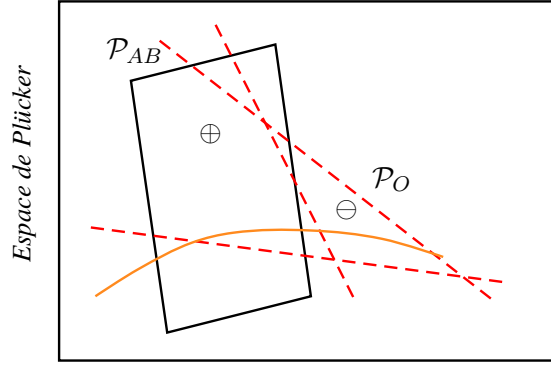


FIG. 2.2 – \mathcal{P}_{AB} et \mathcal{P}_O possèdent une intersection non vide, mais distincte de la quadrique de Plücker. L'opération de soustraction est donc inutile.

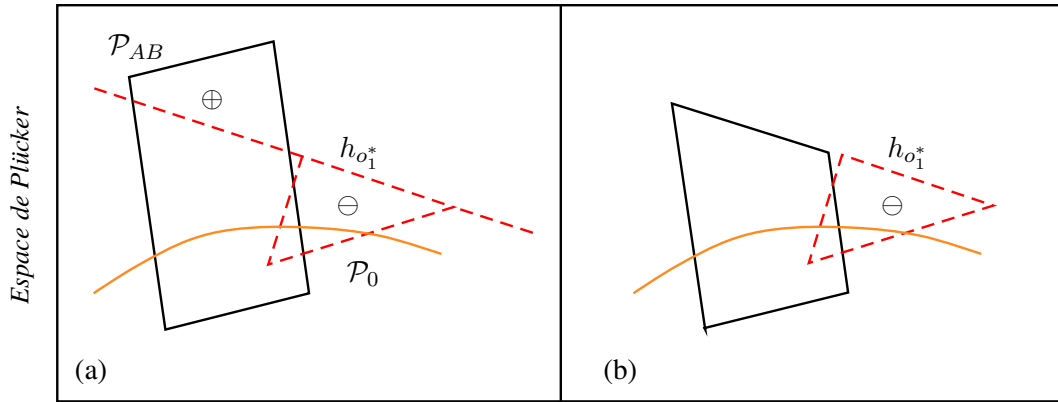


FIG. 2.3 – (a) \mathcal{P}_{AB} et \mathcal{P}_O ont une intersection non nulle, y compris en terme de visibilité. (b) Cependant tous les hyperplans de la H-représentation de \mathcal{P}_O ne donnent pas lieu à des opérations utiles. Ainsi l'intersection par $h_{o_1^*}$ n'affecte pas l'intersection de \mathcal{P}_{AB} avec la quadrique de Plücker. De plus, elle crée un polytope disjoint de la quadrique de Plücker qui peut être supprimé puisqu'il ne représente aucune droite réelle.

La seconde raison qui conduit à effectuer une opération inutile tient dans la difficulté d'effectuer un test d'intersection efficace entre un polytope et un polyèdre. Par propriété de convexité, si deux polytopes sont disjoints, il existe toujours au moins un hyperplan h , élément de la H-représentation d'un polytope, qui rejette tous les sommets de la V-représentation de l'autre polytope (*i.e.* ceux-ci seront tous contenus dans un même demi-espace induit par un hyperplan h). Ce test suppose donc de connaître les H et V-représentations des deux polytopes. Dans le cas d'un polyèdre \mathcal{P}_O et d'un polytope \mathcal{P}_{AB} , ce test n'est que partiellement réalisable. En effet, on ne connaît pas de V-représentation pour \mathcal{P}_O . Plus précisément, \mathcal{P}_O

n'en possède pas car il n'est pas fermé dans l'espace de Plücker. Aussi seul le rejet de la V-représentation d'un polytope par rapport à la H-représentation d'un polyèdre est réalisable. La figure 2.4 en donne une illustration. Ce test partiel est effectué par Nirenstein afin de limiter les calculs d'intersection dans la zone du polyèdre. Cependant le test échoue dans le cas où tous les hyperplans de la H-représentation de \mathcal{P}_O intersectent la V-représentation de \mathcal{P}_{AB} . La figure 2.5.(a) illustre cette configuration.

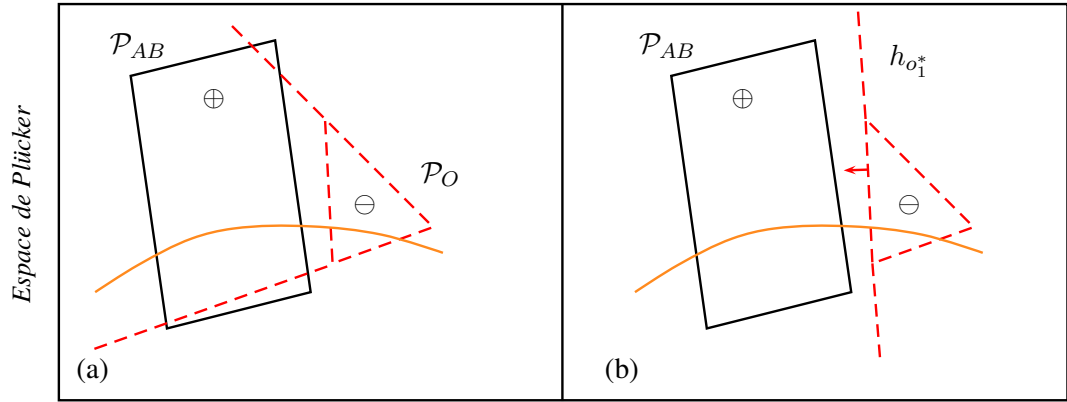


FIG. 2.4 – Rejet de la V-représentation d'un polytope par la H-représentation d'un polyèdre. (a) \mathcal{P}_{AB} et \mathcal{P}_O ne possèdent pas d'intersection, mais 2 des 3 hyperplans de \mathcal{P}_O intersectent \mathcal{P}_{AB} . Le test de rejet vérifie si \mathcal{P}_{AB} est inclus dans le demi-espace positif induit par les hyperplans de \mathcal{P}_O . Des intersections inutiles sont évitées.

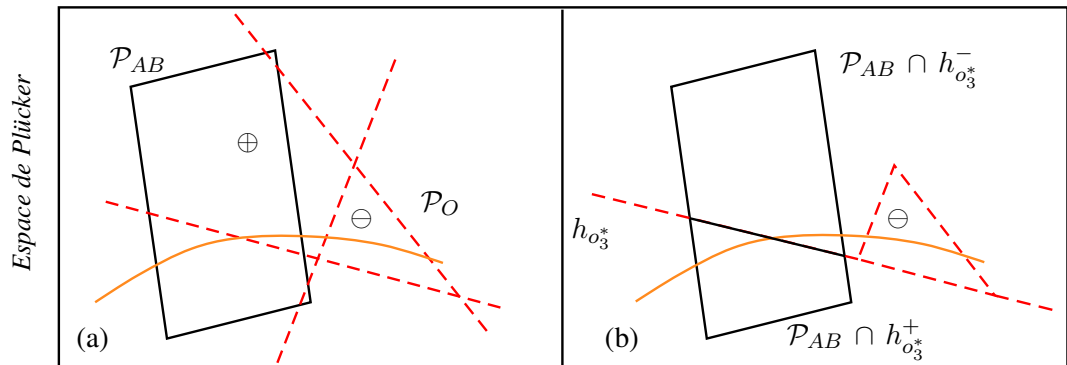


FIG. 2.5 – (a) \mathcal{P}_{AB} et \mathcal{P}_O ne possèdent pas d'intersection commune. Mais tous les hyperplans de \mathcal{P}_O intersectent \mathcal{P}_{AB} . Le test de rejet échoue, et les intersections, bien qu'inutiles, sont calculées. (b) Une conséquence possible est une fragmentation de la visibilité. L'ensemble de visibilité que \mathcal{P}_{AB} suffisait à représenter est maintenant subdivisé en 2 polytopes $\mathcal{P}_{AB} \cap h_{o3}^-$ et $\mathcal{P}_{AB} \cap h_{o3}^+$.

Ainsi, bien que \mathcal{P}_{AB} et \mathcal{P}_O soient disjoints, des intersections seront calculées. Le cas échéant, elles peuvent engendrer une *fragmentation de la visibilité*. Ce "phénomène" est illustré par la figure 2.5.(b) : si une opération inutile subdivise l'intersection d'un polytope avec la quadrique de Plücker, de l'information superflue est créée, *i.e.* deux polytopes au lieu d'un vont être utilisés pour représenter le même ensemble de visibilité.

Notons enfin que l'algorithme de Bittner est d'autant plus sensible à ces problèmes qu'il n'autorise que des tests d'intersection hyperplan/polytope (voir 1.4.10). Par conséquent, aucun test de rejet n'est possible. Il en est de même pour l'algorithme utilisé par Pu *et al.* (voir 1.4.8).

Les exemples qui viennent d'être présentés illustrent le manque de robustesse du test d'intersection polytope/polyèdre et ses conséquences en termes de temps de calcul et de coût mémoire. Des essais préliminaires, utilisant le même test que Nirenstein, ont indiqué que 40 à 60% des intersections calculées s'avéraient inutiles. Pour réduire ce surcoût, il est nécessaire d'améliorer l'efficacité des tests d'intersection.

Pour compléter le test de rejet de \mathcal{P}_{AB} par la H-représentation de \mathcal{P}_O , la V-représentation du polytope P_{A0} ou bien P_{OB} pourrait être calculée pour tester le rejet du bloqueur par la H-représentation de \mathcal{P}_{AB} . C'est une solution contraignante sur laquelle nous reviendrons dans la prochaine section. Mais la connaissance des V-représentations des deux ensembles ne permettrait pas d'éviter le cas d'une intersection non vide mais disjointe de la quadrique de Plücker (voir la figure 2.2). Nous proposons une solution alternative dans la section suivante.

2.2 Approche proposée

Dans cette partie, nous renforçons le test d'intersection d'un polytope et d'un polyèdre en définissant l'*esl-représentation* d'un bloqueur. La finalité est la réduction du nombre d'intersections inutiles, et donc des effets de fragmentation. Puis, nous présentons l'algorithme mis en œuvre pour calculer et stocker la visibilité de deux polygones. Celui-ci intègre le test d'intersection renforcé et propose une synthèse des travaux de Nirenstein et de Bittner.

2.2.1 Test d'intersection renforcé

Nous conservons ici les notations utilisées dans la partie précédente. Comme nous l'avons indiqué, la faiblesse du test d'intersection d'un polytope \mathcal{P}_{AB} et d'un polyèdre \mathcal{P}_O s'explique par l'absence de V-représentation pour ce dernier. Une possibilité évoquée est l'utilisation du polytope \mathcal{P}_{AO} ou \mathcal{P}_{OB} à la place de \mathcal{P}_O . Ces deux polytopes représentent un sous-ensemble suffisant des droites réelles contenues par \mathcal{P}_O . Leurs V-représentations peuvent être calculées de manière analogue à celle de \mathcal{P}_{AB} . Cependant ce choix n'est pas judicieux.

D'une part, il nécessite d'énumérer les sommets d'un polytope pour chaque bloqueur soustrait. D'autre part, il ne permet pas d'utiliser l'algorithme d'énumération proposé par Haumont *et al.* [HMN05]. En effet, celui-ci utilise un hyperplan de projection pour passer de \mathbb{P}^5 à \mathbb{R}^5 . Pour que les polytopes soient comparables, le même hyperplan de projection doit être considéré pour chacun d'entre eux. Rappelons que celui choisi par Haumont est fonction de la droite à l'intersection des plans supports de A et de B (voir 1.4.11). De fait, un hyperplan de projection commun impliquerait que les plans supports des polygones A , B et de tous les bloqueurs O s'intersectent en une seule et même droite. Si cette configuration est possible, elle relève du cas particulier. Aussi, la V-représentation de \mathcal{P}_{AO} (ou \mathcal{P}_{OB}) devrait être calculée avec un algorithme d'énumération non dédié tel que [FR94], moins performant et moins robuste. Pour ces raisons, cette alternative au polyèdre \mathcal{P}_O s'avère coûteuse et contraignante sur le plan algorithmique. Aussi nous proposons une autre solution.

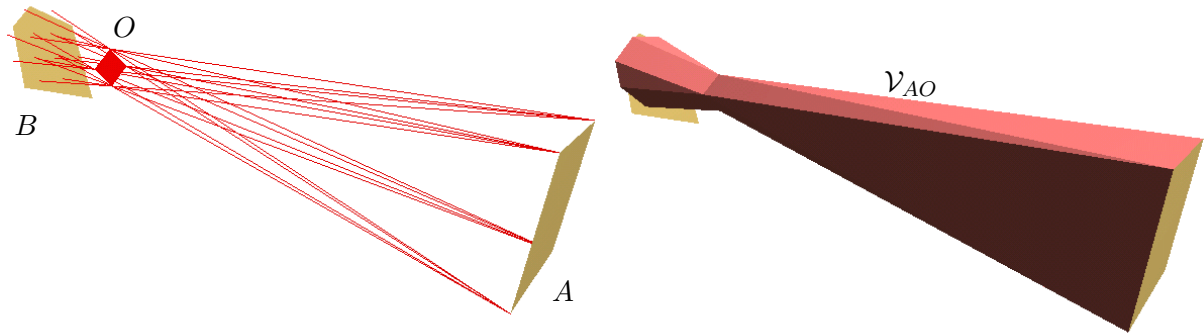


FIG. 2.6 – Les *esl* engendrées par les sommets de A et de O (à gauche) sont les droites supports des arêtes du polyèdre \mathcal{V}_{AO} (à droite). Ce polyèdre délimite l'ensemble des droites qui intersectent A et O . Assimilant A à une source lumineuse, \mathcal{V}_{AO} représente l'antipénombre de A à travers O .

Considérons l'ensemble des droites intersectant A et O . Dans \mathbb{R}^3 , cet ensemble est contenu dans un polyèdre (non convexe) ouvert dans deux directions. Ses demi-arêtes ont pour droites supports les *esl* générées par les sommets de A et les sommets de O . Il peut être vu comme délimitant l'antipénombre de A à travers O en assimilant A à une source lumineuse. Notons \mathcal{V}_{AO} ce polyèdre de \mathbb{R}^3 dont la figure 2.6

donne une illustration.

Une droite l qui partitionne (au sens de Plücker) l'ensemble des droites contenues dans \mathcal{V}_{AO} intersecte ce polyèdre. De fait, dans l'espace de Plücker, si les esl générées par A et O sont toutes contenues dans un même demi-espace induit par un hyperplan h_{l^*} , cela signifie que la droite l n'intersecte pas \mathcal{V}_{AO} , et donc qu'elle ne partitionne pas les droites qu'il représente. La figure 2.7 illustre cette propriété qui peut être utilisée pour effectuer un test de rejet sur un bloqueur :

- Si la H-représentation d'un polytope \mathcal{P} rejette les esl générées par A et O alors il n'existe pas de droite réelle commune à \mathcal{P} et à \mathcal{V}_{AO} .

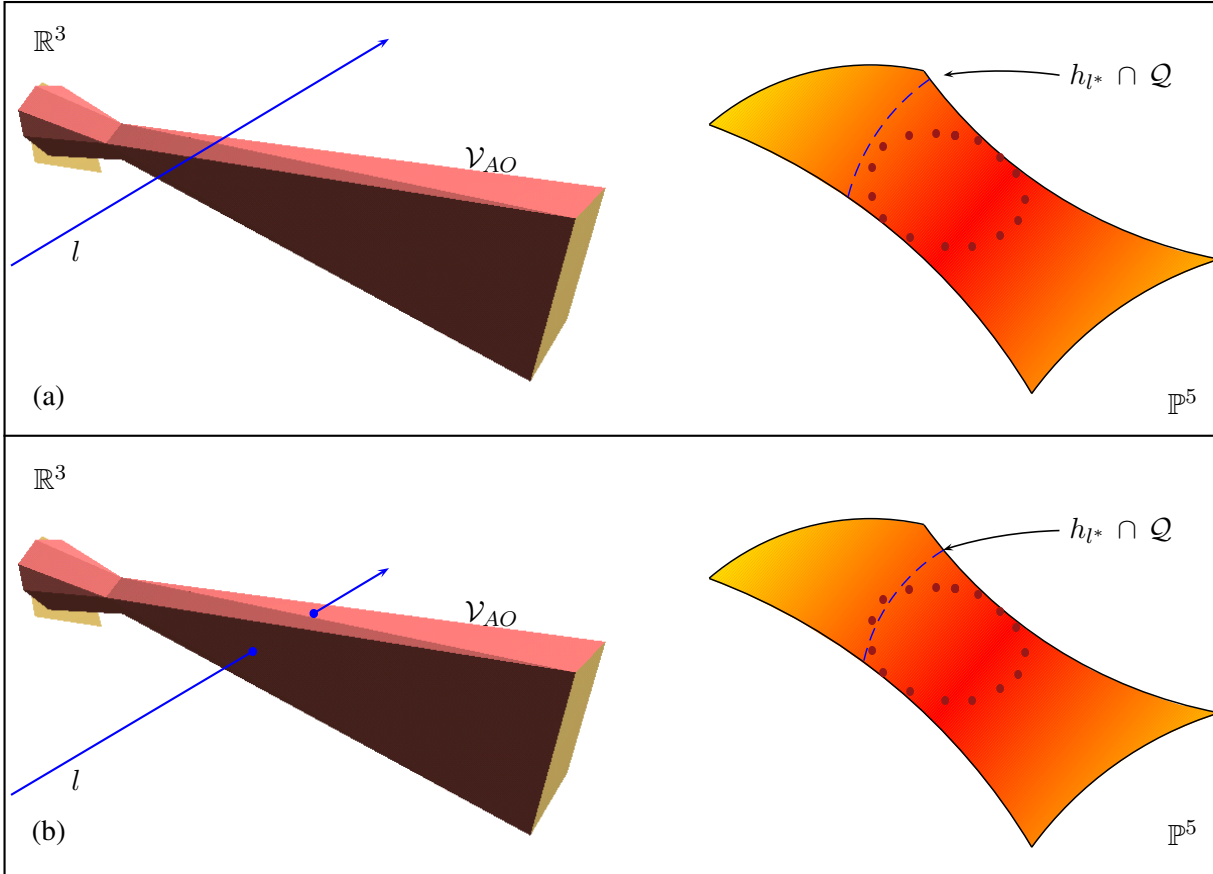


FIG. 2.7 – Pour les deux schémas ci-dessus : à gauche, une droite l et le polyèdre \mathcal{V}_{AO} (défini par deux quadrilatères) dans \mathbb{R}^3 ; à droite dans l'espace de Plücker, les 16 esl , droites supports des arêtes de \mathcal{V}_{AO} , et la trace de l'intersection de l'hyperplan h_{l^*} avec la quadrique de Plücker \mathcal{Q} . (a) Si l n'intersecte pas \mathcal{V}_{AO} , l'ensemble des esl est contenu dans le même demi-espace induit par h_{l^*} . (b) Si l intersecte \mathcal{V}_{AO} , l'hyperplan h_{l^*} partitionne l'ensemble des esl .

Jusqu'à présent, nous avons raisonné arbitrairement avec \mathcal{V}_{AO} . Bien entendu, \mathcal{V}_{BO} est également une représentation valide pour appliquer le test de rejet au bloqueur O . Par souci d'efficacité, il convient de choisir la plus petite des deux représentations (par conséquent celle qui s'approche le plus de l'ensemble des droites intersectant A , O , et B). En pratique, nous calculons une approximation des angles solides [CT96, CT97] de A et de B depuis le barycentre du bloqueur O . Le polygone retenu est celui associé au plus petit angle solide. L'ensemble des esl résultantes forme ce que nous appelons l' esl -représentation d'un bloqueur. L'algorithme 1 présente le pseudo-code pour son calcul.

Algorithme 1 : Pseudo-code pour le calcul de l'esl-représentation d'un bloqueur. La fonction *EslRepresentation* (ligne 1) retourne les *esl* qui seront utilisées pour tester le rejet du bloqueur O par la H-représentation d'un polytope représentatif de visibilité. Une approximation des angles solides (ligne 9) de A et B depuis le barycentre du bloqueur O est calculée. La plus petite valeur détermine avec quel polygone les *esl* seront calculées (ligne 14-19).

```

1 fonction EslRepresentation (Polygone  $A, B, O$ ) retourne EslRep
2 Point3D  $p$ 
3 Polygone candidat
4 EslRep envOccluder
5 Plucker droite
6 début
7   envOccluder  $\leftarrow \emptyset$ 
8    $p \leftarrow \text{barycentre}(O)$ 
9   si AngleSolide( $p, A$ ) < AngleSolide( $p, B$ ) alors
10    | candidat  $\leftarrow A$ 
11  sinon
12    | candidat  $\leftarrow B$ 
13  fin
14  pour chaque sommet  $v_i$  de candidat faire
15    | pour chaque sommet  $v_j$  de  $O$  faire
16      | | droite  $\leftarrow \text{Map2Plucker}(v_i, v_j)$ 
17      | | envOccluder  $\leftarrow \text{envOccluder} \cup \text{droite}$ 
18    | fin
19  fin
20  retourner envOccluder
21 fin

```

Notons que ce test est simple et peu coûteux. Le calcul de l'esl-représentation a pour complexité $O(nm)$ (où n et m sont respectivement les nombres de sommets pour A (ou B) et O). De plus, il présente l'avantage de s'appliquer sur l'ensemble des droites réelles intersectant A et O et non sur une représentation plus grande de cet ensemble (telle que \mathcal{P}_{AO} par exemple). À ce sujet, on peut objecter que l'ensemble des droites intersectant A et O est supérieur ou égal à l'ensemble des droites intersectant A , O et B . Trois polygones en position générale sont générateurs d'événements EEE. Les surfaces associées à ces événements sont des surfaces réglées dans \mathbb{R}^3 . Aussi les droites intersectant trois polygones ne sont pas contenues dans un polyèdre de \mathbb{R}^3 . Dans ce cas, le test de rejet sur l'esl-représentation du bloqueur n'est plus valide. Il est ainsi nécessaire de se restreindre à deux polygones.

L'esl-représentation d'un bloqueur qui vient d'être présentée permet de compléter le test d'intersection d'un polytope et d'un polyèdre. Dans un contexte d'illumination globale, il est à noter qu'implicitement, Teller et Hanrahan [TH93] utilisent un test similaire comme critère de subdivision pour le calcul d'un maillage de polygones.

2.2.2 Arbre V BSP

Cette section présente notre approche pour le calcul d'une représentation cohérente de la visibilité entre deux polygones convexes. Celle-ci intègre le test renforcé de l'intersection d'un bloqueur et d'un polytope et propose une synthèse des travaux de Nirenstein et de Bittner. Du premier, nous retenons l'efficacité des algorithmes utilisés, d'une part, pour l'énumération de la V-représentation d'un polytope, et d'autre part, pour les opérations CSG. Du second, nous retenons la structure d'arbre BSP dans l'espace de Plücker pour obtenir une représentation hiérarchique de la visibilité.

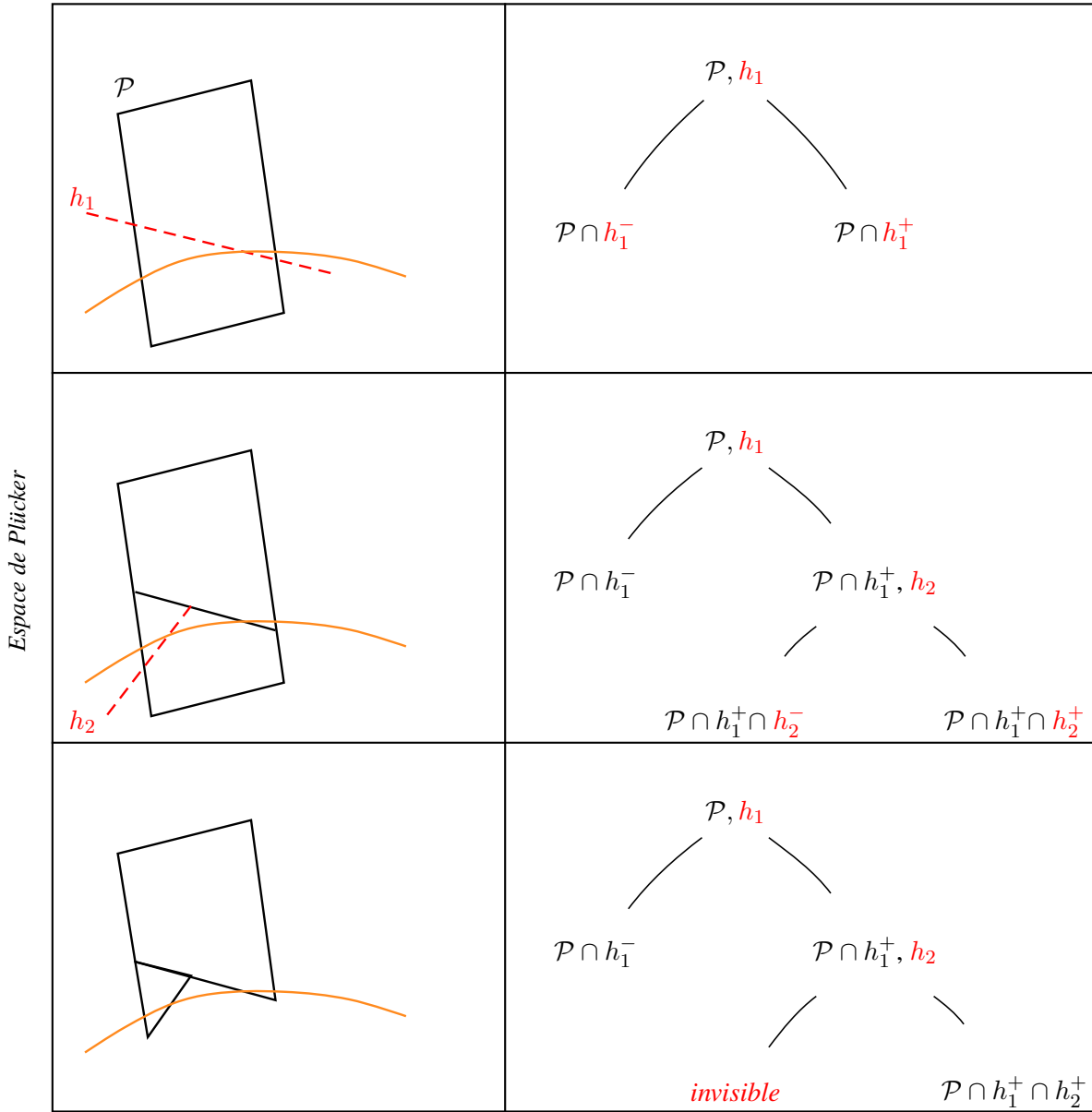


FIG. 2.8 – Partition d'un ensemble de visibilité. Si un bloqueur possède une intersection avec le polytope contenu dans une feuille *visible*, celle-ci est remplacée par le sous-arbre représentatif de la partition de \mathcal{P} par la H-représentation du bloqueur (ici $\{h_1, h_2\}$).

Cependant, l'algorithme de Bittner présente certains écueils. Il nécessite l'énumération des sommets d'un polytope pour chaque bloqueur traité, ce qui est coûteux. L'insertion d'un polytope dans l'arbre rend les calculs sensibles à la zone des hyperplans présents dans les noeuds internes, ce qui n'est pas souhaitable en terme de cohérence de l'information calculée. Nous proposons un algorithme différent qui permet d'éviter ces problèmes. Le résultat est un arbre BSP dans l'espace de Plücker représentatif de la visibilité de deux polygones. De plus, ne possédant pas d'information de profondeur, il diffère également d'un arbre d'occultation sur l'information représentée. Nous le nommons arbre VBSP.

Algorithme 2 : Pseudo-code simplifié de la construction d'un arbre VBSP.

```

22 fonction Initialiser (Polygone  $A, B$ ) retourne VBSP
23   Polytope polytope
24   VBSP arbre
25   début
26     polytope  $\leftarrow$  creer_polytope ( $A, B$ )
27     arbre  $\leftarrow$  creer_feuille (polytope , visible)
28     retourner arbre
29   fin

30 procedure inserer_bloqueur (VBSP arbre, Bloqueur  $O$ )
31   début
32     si rejet_echec ( $O.hrep$ , arbre.polytope) alors
33       si arbre.type = feuille et arbre.label = visible alors
34         | Partitionner (arbre,  $O$ )
35       sinon
36         suivant (arbre.hyperplan  $\cap$   $O.eslrep$ ) faire
37           cas où POSITIF
38             | inserer_bloqueur (arbre.posarbre,  $O$ )
39           cas où NEGATIF
40             | inserer_bloqueur (arbre.negarbre,  $O$ )
41           autres cas
42             | inserer_bloqueur (arbre.posarbre,  $O$ )
43             | inserer_bloqueur (arbre.negarbre,  $O$ )
44           fin
45         fin
46       fin
47   fin

48 fonction VisibilitePoly2Poly (Polygone  $A, B$ ) retourne VBSP
49   VBSP arbre
50   début
51     arbre  $\leftarrow$  Initialiser ( $A, B$ )
52     pour chaque bloqueur  $O$  de  $A$  et  $B$  faire
53       |  $O.eslrep$   $\leftarrow$  EslRepresentation ( $A, B, O$ )
54       |  $O.hrep$   $\leftarrow$  HRepresentation ( $O$ )
55       | inserer_bloqueur (arbre,  $O$ )
56     fin
57     retourner arbre
58   fin

```

Un arbre VBSP ne contient pas d'information de profondeur. Ceci permet de fusionner systématiquement tout ensemble de droites bloquées et de traiter les bloqueurs dans n'importe quel ordre donné. Nous pouvons tirer avantage du mécanisme d'ordonnancement des bloqueurs (voir 1.4.9) privilégié par Nirenstein ou par Haumont, ce qui n'est pas possible pour un arbre d'occultation. Chaque nœud interne n_i d'un arbre VBSP contient, en plus d'un hyperplan, un polytope \mathcal{P}_i qui est la région de l'espace partitionnée par le sous-arbre enraciné en n_i . Chaque feuille est marquée *invisible* ou *visible*. Dans ce dernier cas, la feuille contient un polytope représentatif de l'ensemble de visibilité.

Le principe de l'algorithme est de tester hiérarchiquement l'intersection des polytopes avec chaque blo-

queur, et de reporter tous les calculs d'intersections dans les feuilles de l'arbre VBSP. Lorsqu'une feuille f_i *visible* est atteinte, l'algorithme effectue la soustraction du bloqueur au polytope \mathcal{P}_i contenu. Le sous-arbre représentatif de la partition de \mathcal{P}_i par la H-représentation du bloqueur O , est d'abord calculé, puis enraciné en f_i . La figure 2.8 illustre la construction d'un tel sous-arbre. Cet algorithme permet de minimiser la fragmentation de la visibilité. D'une part, les intersections calculées sont locales aux polytopes contenus dans les feuilles *visibles*, et ne sont donc pas sensibles à la zone des hyperplans rencontrés lors de l'insertion du bloqueur. D'autre part, ces polytopes sont les plus petits représentants des visibilité restantes. Comme nous le verrons dans la partie suivante, les risques d'effectuer des intersections non pertinentes s'en trouvent significativement réduits.

L'algorithme 2 donne le pseudo-code simplifié de la construction d'un arbre VBSP. Celui-ci est initialisé (ligne 52) avec une feuille marquée *visible* contenant le polytope représentant l'ensemble des droites intersectant deux polygones convexes A et B . Les bloqueurs pour A et B résultent de l'intersection de la géométrie de la scène avec l'enveloppe convexe de A et B . Pour chaque bloqueur O , son esl-représentation (ligne 54) et sa H-représentation (ligne 55) sont calculées. Seules les arêtes strictement contenues dans l'enveloppe convexe de A et B participent à la H-représentation d'un bloqueur. Ensuite, le bloqueur est inséré dans l'arbre VBSP (ligne 56). La H-représentation d'un bloqueur est utilisée pour tester le rejet du polytope associé à un noeud (ligne 32). Si le polytope est rejeté, le polytope et le bloqueur sont disjoints et il n'y a pas lieu de poursuivre avec ce bloqueur pour cet arbre VBSP. Sinon, si une feuille *visible* est atteinte (ligne 33), elle est remplacée par le sous-arbre associé à la partition du polytope par le bloqueur O (ligne 34). S'il s'agit d'un noeud interne, l'esl-représentation du bloqueur est testée par rapport à l'hyperplan contenu (ligne 36). Si toutes les esl sont rejetées du côté positif ou négatif de l'hyperplan, l'algorithme se poursuit dans le sous-arbre approprié (lignes 37-40). Si elles sont réparties de part et d'autre de l'hyperplan, on ne peut pas statuer, et l'algorithme se poursuit dans les deux sous-arbres (lignes 41-43).

Lorsqu'une feuille *visible* est atteinte, c'est que le polytope associé et le bloqueur courant n'ont pu être déterminés comme étant disjoints. Cependant, comme indiqué dans la section 2.2.1, cela n'implique pas nécessairement l'intersection des droites réelles qu'ils représentent. Il faut donc vérifier la pertinence des opérations effectuées. La section suivante indique comment notre approche rentabilise les calculs inutiles qui n'ont pu être évités.

2.2.3 Valider et rentabiliser

Bien que renforcé, le test d'intersection polytope/polyèdre n'est pas exact, dans la mesure où il n'est pas précisément effectué sur les ensembles de droites réelles. Le cas échéant, des intersections inutiles peuvent encore être effectuées. Puisqu'elles ne peuvent toutes être évitées, nous proposons ici de les rentabiliser.

Chaque fois que l'intersection d'un hyperplan h et d'un polytope \mathcal{P} est calculée, il faut vérifier la pertinence de l'opération : est-ce que les deux polytopes \mathcal{P}^- et \mathcal{P}^+ résultants possèdent encore une intersection avec la quadrique de Plücker ? L'intersection est inutile à chaque fois que la réponse est non pour l'un des deux polytopes. La stratégie adoptée consiste alors à annuler l'opération tout en conservant le plus petit représentant (au sens de l'inclusion) de l'ensemble de visibilité incriminé. La figure 2.9 illustre ce cas de figure. En travaillant "au plus près" de la quadrique de Plücker, on ne peut qu'améliorer à la fois l'efficacité des futurs tests d'intersection, ainsi que celle des opérations réalisées.

La stratégie précédente s'applique après chaque intersection polytope/hyperplan. Après chaque soustraction d'un bloqueur, les feuilles de l'arbre sont examinées afin de fusionner les ensembles de droites dès

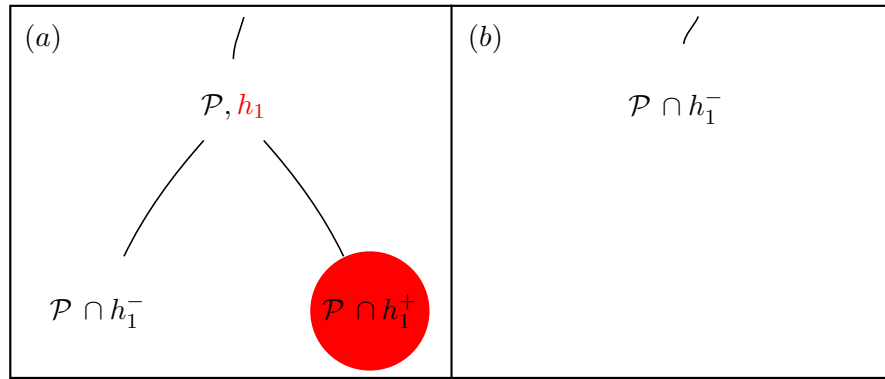


FIG. 2.9 – Rentabiliser les opérations inutiles. (a) \mathcal{P} est intersecté par h_1 . Le produit de l'intersection est $\mathcal{P} \cap h_1^-$ et $\mathcal{P} \cap h_1^+$. Si $\mathcal{P} \cap h_1^+$ (par exemple) ne possède pas d'intersection avec la quadrique de Plücker, l'opération était inutile. $\mathcal{P} \cap h_1^-$ représente le même ensemble de droites réelles que \mathcal{P} . (b) Dans ce cas, l'intersection est annulée mais le polytope \mathcal{P} est remplacé par $\mathcal{P} \cap h_1^-$ qui est un plus petit représentant au sens de l'inclusion (*i.e.* $(\mathcal{P} \cap h_1^-) \subset \mathcal{P}$).

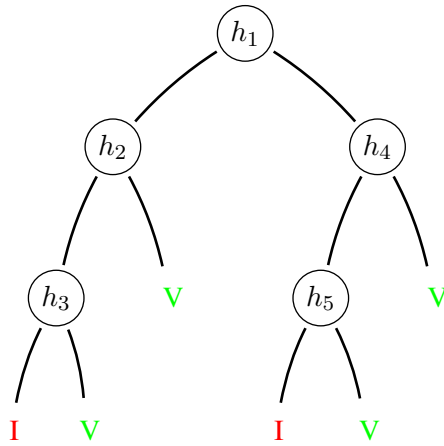


FIG. 2.10 – Après soustraction de tous les bloqueurs, les polytopes contenus dans les nœuds et feuilles de l'arbre VBSP peuvent être supprimés, pour ne conserver que la représentation BSP de la visibilité des deux polygones.

que possible. Tout nœud possédant deux feuilles *invisibles* (resp. *visibles*) devient une feuille *invisible* (resp. *visible*). Dans le cas de deux ensembles de visibilité, le polytope associé à leur père commun est le nouveau représentant. Cette fusion systématique permet de regrouper des ensembles de visibilité inutilement fragmentés.

Une fois que le calcul de la visibilité entre deux polygones A et B est terminé, ce qui nous intéresse de conserver est la représentation de leur visibilité. Dans ce cadre, il n'est pas utile de garder les polytopes associés à chaque nœud de l'arbre VBSP. Ainsi, à la fin de l'algorithme, ils sont tous supprimés et seule demeure la structure "classique" d'arbre BSP, comme l'illustre la figure 2.10.

2.3 Implantation et évaluation

Cette partie présente tout d'abord une vue d'ensemble de notre implantation. Puis elle présente les résultats obtenus sur différentes scènes de tests, et évalue la complexité pratique de notre approche. Enfin sont discutées les améliorations possibles et les développements à donner à ce travail.

2.3.1 Éléments d'implantation

L'implantation du calcul de la visibilité dans l'espace de Plücker fait appel à différents algorithmes non triviaux. Nous précisons ici ceux utilisés dans notre implantation en justifiant nos choix. En particulier, nous expliquons l'optimisation qui peut être réalisée pour calculer le graphe d'incidence d'un polytope. Trois étapes peuvent être distinguées :

- Le calcul de la V-représentation du polytope \mathcal{P}_{AB} associé à deux polygones A et B , ainsi que son graphe d'incidences.
- La sélection et l'ordonnancement des bloqueurs potentiels pour A et B .
- Et enfin, la construction d'un arbre VBSP à proprement parler.

Première étape : V-représentation et graphe d'incidence

La V-représentation d'un polytope est calculée selon l'algorithme proposé par Nirenstein [Nir03] (voir aussi 1.4.9). Puis vient le calcul de son graphe d'incidence. Il s'agit d'une étape coûteuse. C'est l'une des raisons pour lesquelles Haumont [HMN05] développe une approche n'utilisant que le 1-squelette du polytope, quitte à introduire par la suite des opérations numériques supplémentaires. Notre implantation calcule un graphe d'incidence complet en utilisant l'algorithme de Kaibel *et al.* [KP02] contrairement à Nirenstein qui utilise une version similaire à [FR94]. Ce choix se justifie sur le plan des complexités.

En effet, l'algorithme de Fukuda *et al.* [FR94] possède une complexité en $O(\min(s, f) \alpha \varphi^2)$, où s est le nombre de sommets, f le nombre de facettes, α le nombre d'incidences sommets-facettes, et φ le nombre total de faces du polytope ².

Notre implantation utilise l'algorithme de Kaibel *et al.* [KP02], qui a dans le cas général une complexité en $O(\min(s, f) \alpha \varphi)$. De plus, dans notre contexte, le polytope associé à deux polygones est simple (voir 1.2.3). Dans un tel cas de figure, l'algorithme peut être spécialisé en $O(d \alpha \varphi)$, d étant la dimension du polytope. En outre, si plusieurs requêtes doivent être effectuées, il est possible de s'affranchir du coût inhérent à cette étape.

En effet, un graphe d'incidence n'a pas besoin d'être calculé pour chaque requête. Il s'agit d'une information topologique qui dépend uniquement des relations d'incidence sommets-hyperplans du polytope, et aucunement de la position géométrique de ses sommets dans l'espace. Par exemple, le graphe d'incidence d'un cube dans \mathbb{R}^3 est le graphe d'incidence de n'importe quel cube. S'agissant d'un polytope représentant les droites intersectant deux polygones A et B , on peut observer que les relations d'incidences sommets-hyperplans ne dépendent que du nombre de sommets de A et du nombre de sommets de B .

Notons $\{a_1 \dots a_i \dots a_n\}$ les arêtes de A et $\{b_1 \dots b_j \dots b_m\}$ celles de B . Les droites reliant chaque sommet de A à chaque sommet de B sont les *esl* notées s_{ij} dans l'espace de Plücker. Chaque s_{ij}^* est incidente aux hyperplans $\{h_{a_i}^*, h_{a_{i+1}}^*, h_{b_j}^*, h_{b_{j+1}}^*\}$. Par projection sur les hyperplans H^+ et H^- bornant

²A titre indicatif : Pour deux triangles $(s, f, \alpha, \varphi) = (18, 8, 40, 147)$. Pour deux quadrilatères $(s, f, \alpha, \varphi) = (32, 10, 160, 243)$

le polytope, chaque *esl* donne 2 sommets du polytope s_{ij}^{*+} et s_{ij}^{*-} , respectivement incidents aux hyperplans $\{H^+, h_{a_i}^*, h_{a_{i+1}}^*, h_{b_j}^*, h_{b_{j+1}}^*\}$ et $\{H^-, h_{a_i}^*, h_{a_{i+1}}^*, h_{b_j}^*, h_{b_{j+1}}^*\}$. Ainsi les relations d'incidence sommets-hyperplans d'un polytope sont uniquement fonction du nombre de sommets de A et du nombre de sommets de B .

Par conséquent, pour un couple de polygones à n et m sommets, le graphe d'incidence du polytope associé est calculé une seule fois. Ce graphe est ensuite réutilisé pour tous les autres couples similaires (en terme de nombres de sommets) en mettant à jour les plongements géométriques (*i.e.* les coordonnées des sommets). Par exemple pour une scène triangulée, un seul graphe d'incidence a besoin d'être calculé. Notre implantation s'affranchit ainsi du temps consacré au calcul des graphes d'incidence.

Deuxième étape : sélection ordonnée des bloqueurs

La sélection des bloqueurs pour A et B nécessite d'intersecter la géométrie de la scène avec l'enveloppe convexe de A et B . Pour casser la complexité linéaire de cette étape vis à vis du nombre de polygones dans la scène, les calculs d'intersections sont hiérarchisés en se basant sur une subdivision spatiale de la scène (une hiérarchie de boîtes englobantes dans le cas présent).

Une première sélection conservative est effectuée en utilisant le *shaft* [Hai00] des deux polygones A et B , *i.e.* l'enveloppe convexe des boîtes englobantes de A et B . La subdivision spatiale est traversée par le *shaft* afin de sélectionner les feuilles qui l'intersectent. Il en résulte un premier ensemble conservatif de bloqueurs pour A et B . L'intersection de cet ensemble avec l'enveloppe convexe de A et B procure l'ensemble exact des bloqueurs. Ils sont ensuite ordonnancés selon le mécanisme proposé par Nirenstein et résumé en 1.4.9.

Troisième étape : construction de l'arbre VBSP

La troisième et dernière étape concerne le calcul de l'arbre VBSP. Toutes les intersections hyperplans/polytopes sont effectuées sur la base de l'algorithme de Bajaj et Pascucci [BP96]. Les bloqueurs sont traités successivement jusqu'à terminaison de l'algorithme : soit A et B sont invisibles (*i.e.* il ne reste plus de feuilles visibles), soit il ne reste plus de bloqueur à traiter. Dans ce dernier cas, l'arbre VBSP résultant et représentatif de la visibilité de A et B est conservé.

2.3.2 Évaluation de la méthode

Pour évaluer³ notre approche, nous comparons d'une part la construction d'un arbre VBSP en utilisant uniquement un test d'intersection partiel (utilisé par Nirenstein), et d'autre par sa construction avec le test renforcé allié à la conservation du plus petit représentant. Pour chacune des requêtes effectuées entre deux polygones, les valeurs suivantes sont mesurées :

- Le temps d'exécution de la requête
- Le nombre d'intersections polytope/hyperplan calculés
- Le nombre final de polytopes représentant la visibilité de deux polygones
- Le nombre de noeuds présents dans l'arbre VBSP

Deux séries de tests sont effectuées. La première série utilise des scènes générées pour confronter notre approche aux configurations les plus difficiles, *i.e.* lorsque la complexité visuelle devient importante. La seconde série utilise une scène urbaine afin d'évaluer la méthode dans un environnement réaliste. Les résultats sont donnés en fonction du nombre de bloqueurs effectifs. Un bloqueur effectif est défini

³Tests effectués sur un AMD64 3200+ (2.2GHz) avec 2Go de mémoire

comme étant élément du plus petit ensemble de bloqueurs définissant l'occultation. En pratique, un bloqueur est effectif s'il a contribué à supprimer une partie (au moins) des visibilitées. Les courbes "test partiel" correspondent à l'utilisation du test d'intersection partiel polytope/polyèdre. De plus, les polytopes sans intersection avec la quadrique de Plücker sont systématiquement supprimés, mais les opérations inutiles ne sont pas rentabilisées. Les courbes "vbsp" correspondent à notre approche basée sur le test d'intersection renforcé et sur la rentabilisation des calculs inutiles.

Environnement généré

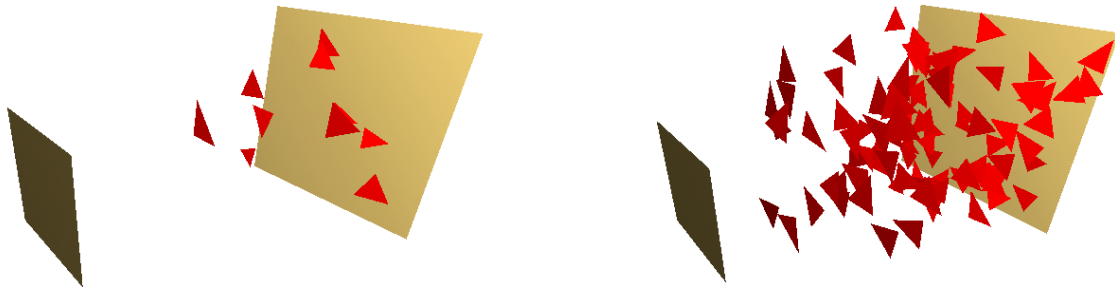


FIG. 2.11 – Deux exemples de la série de scènes. Les deux quadrilatères demeurent visibles mais la complexité visuelle entre eux augmente.

Nous nous intéressons ici à l'évolution du calcul de visibilité entre deux polygones lorsque la complexité visuelle entre eux augmente. Les scènes générées comprennent toutes deux quadrilatères, plus un nombre croissant de bloqueurs aléatoirement répartis. La figure 2.11 en montre deux exemples. Précisons que les deux quadrilatères demeurent toujours visibles dans toutes les scènes générées. Pour chaque scène de test, la visibilité des deux quadrilatères est calculée. Les valeurs présentées sont les moyennes des mesures effectuées sur 5 scènes différentes à n bloqueurs chacune.

Le temps d'exécution d'une requête est proportionnel au nombre d'intersections calculées. En ce sens, les graphes 2.12.a et 2.12.b sont à considérer conjointement. Si le nombre d'intersections est indiqué, c'est qu'il ne dépend que de l'algorithme, contrairement au temps de calcul qui est quant à lui lié à l'efficacité de l'implantation. Les résultats indiquent une réduction de l'ordre de 60% du nombre d'intersections calculées. Le temps d'exécution d'une requête est alors diminué d'autant. Une estimation par les moindres carrés indique une complexité pratique sensiblement inférieure à $O(n^2)$ (précisément, $O(n^{1.99})$ pour 2.12.a et $O(n^{1.92})$ pour 2.12.b, n étant le nombre de bloqueurs).

Le graphe 2.12.c montre le nombre final de polytopes représentant la visibilité des deux quadrilatères, *i.e.* le nombre de feuilles marquées *visibles*. Ceci nous donne un critère permettant d'évaluer la fragmentation de la visibilité. Celle-ci est réduite de 60% par notre approche. Une estimation par les moindres carrés indique une complexité mémoire en $O(n^{1.44})$. Ce résultat est loin d'être aussi prohibitif que la complexité théorique annoncée en $O(n^4 \log n)$. Rappelons que cette dernière est liée à l'arrangement des hyperplans associés aux arêtes des bloqueurs. Les cellules pertinentes (*i.e.* celles qui correspondent à un ensemble de visibilité entre les deux quadrilatères) sont un sous-ensemble de toutes les cellules de l'arrangement. À ce titre, une complexité pratique inférieure était attendue.

Ces polytopes correspondent aux feuilles de l'arbre VBSP marquées visibles. Ils ne sont donc pas représentatifs de toute la mémoire nécessaire au stockage d'un arbre VBSP. Cette dernière est mesurée sur le graphe 2.12.d. Comme précédemment, on retrouve une diminution d'environ 60% du nombre de noeuds.

La complexité mémoire d'un arbre VBSP est estimée en $O(n^{1.76})$.

Cette première étude montre que notre approche permet une diminution globale de 60% des ressources

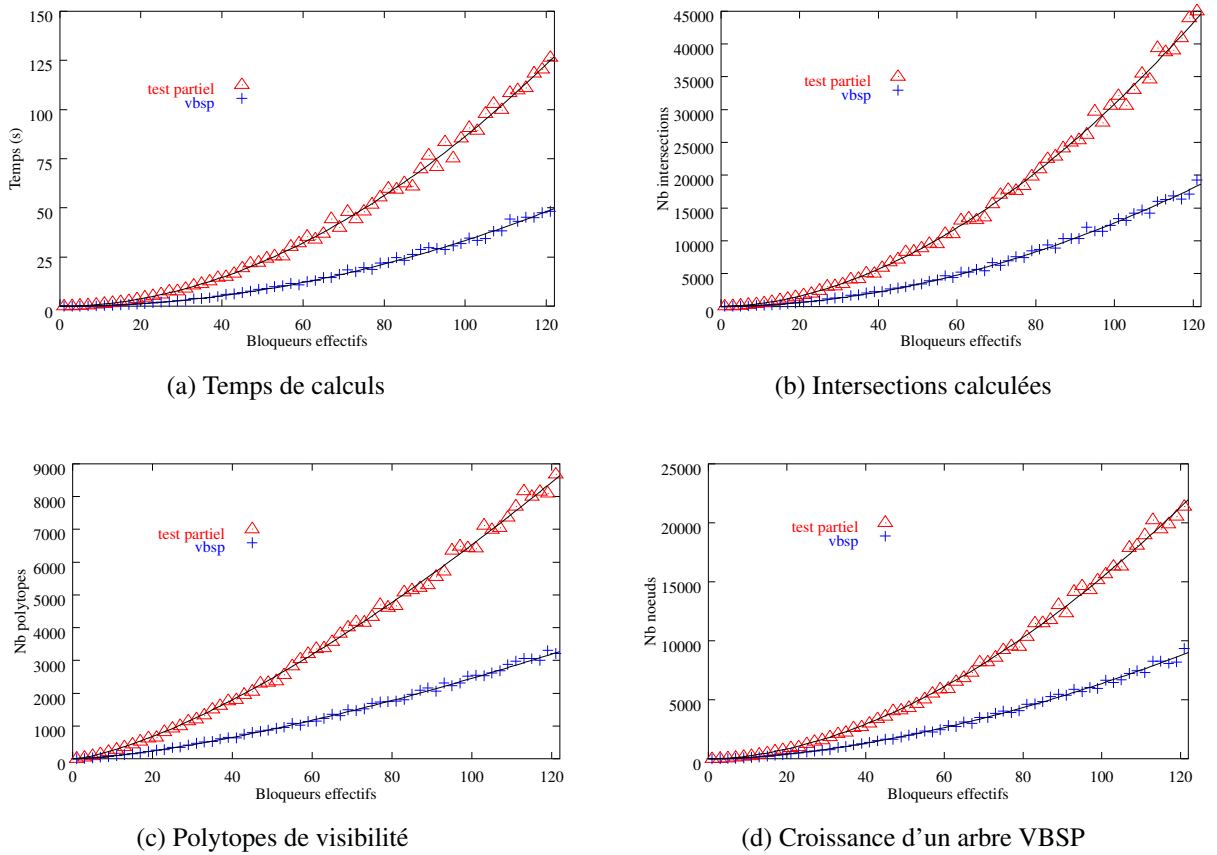


FIG. 2.12 – Résultats obtenus sur la série de scènes tests générées de manière à avoir une complexité visuelle croissante en fonction du nombre de bloqueurs effectifs.

temps et mémoires utilisées lors d'une requête de visibilité entre deux polygones. On peut noter qu'en diminuant le nombre d'intersections calculées, on minimise également le risque d'instabilités numériques. Enfin, bien que les complexités temps et mémoires estimées soient conséquentes, elles restent nettement inférieures à $O(n^4 \log n)$.

Environnement réaliste

Dans cette seconde série de tests, nous nous plaçons dans un environnement réaliste, à savoir un quartier de Paris (figure 2.13). La visibilité est calculée pour chaque couple de polygones de la scène⁴. La figure 2.14 présente l'ensemble des résultats obtenus. De manière générale, deux éléments sont à relever :

En premier, notons que quelles que soient les grandeurs mesurées, la construction VBSP apparaît bornée en temps et en mémoire, alors que la méthode partielle conserve une complexité polynomiale en fonction du nombre de bloqueurs effectivement soustraits. Les graphes 2.14.(a) et 2.14.(b) indiquent que la durée d'une requête n'excède pas 0.9 seconde pour 530 intersections calculées au plus, quand la

⁴Ce type de calcul sera à nouveau utilisé au chapitre 5.

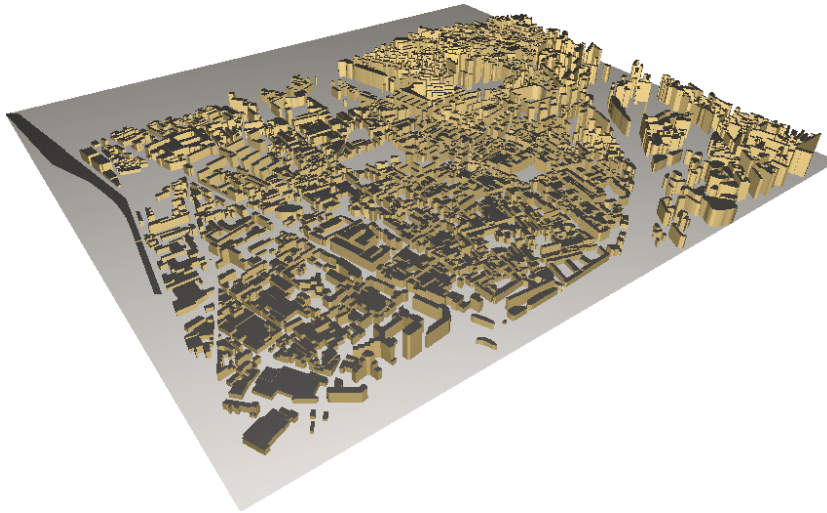


FIG. 2.13 – Aperçu de la scène extérieure utilisée pour la seconde série de tests. La visibilité de chaque couple de polygones est ici évaluée.

méthode partielle atteint 4 secondes et 2000 intersections. Le graphe 2.14.(c) montre que moins de 40 polytopes suffisent à décrire la visibilité alors que la méthode partielle en utilise jusqu'à plus de 450. De même, le graphe 2.14.(d) indique que la taille des arbres demeure inférieure à 160 nœuds dans le premier cas, et dépasse les 1000 dans le second.

Ces résultats illustrent l'impact du test renforcé sur la localisation des calculs à la zone des bloqueurs : à nombre de bloqueurs effectifs égal, notre approche engendre un nombre borné d'intersections. Contrairement à la précédente série de tests, la complexité visuelle n'augmente pas constamment avec le nombre de bloqueurs effectifs, mais est bornée (*i.e.* il existe dans la scène un couple de polygones pour lesquels la complexité visuelle est maximale). Ces résultats indiquent que le comportement de notre approche est corrélée à la complexité visuelle, quand la méthode partielle reste sensible au nombre de bloqueurs effectifs.

Le second élément notable est la différence d'ordre de grandeur avec la précédente série de test. Globalement, les valeurs mesurées ici sont 20 fois plus petites.

Cet écart reflète la différence de complexité visuelle entre les deux séries de tests. Les configurations difficiles que présentent la première n'existent pas dans la seconde, où la complexité visuelle entre deux polygones est moindre. Bien que le contraste soit sans aucun doute accentué par la nature 2.5D de la seconde scène, cela montre aussi que ce type de calcul n'est pas nécessairement prohibitif selon la nature de l'environnement.

Ce constat n'est pas sans rappeler les travaux de Goaoc [Goa04, BDD⁺04]. Rappelons que pour une distribution uniforme d'objets, il montre qu'une approche globale de la visibilité n'est pas aussi dissuasive que les bornes théoriques le laissent entendre dans le cas général (voir aussi 1.4.6).

2.3.3 Discussion

Les résultats précédents montrent que notre approche apporte une réponse au problème de fragmentation de la visibilité. Le test d'intersection renforcé améliore la localisation des opérations CSG à la

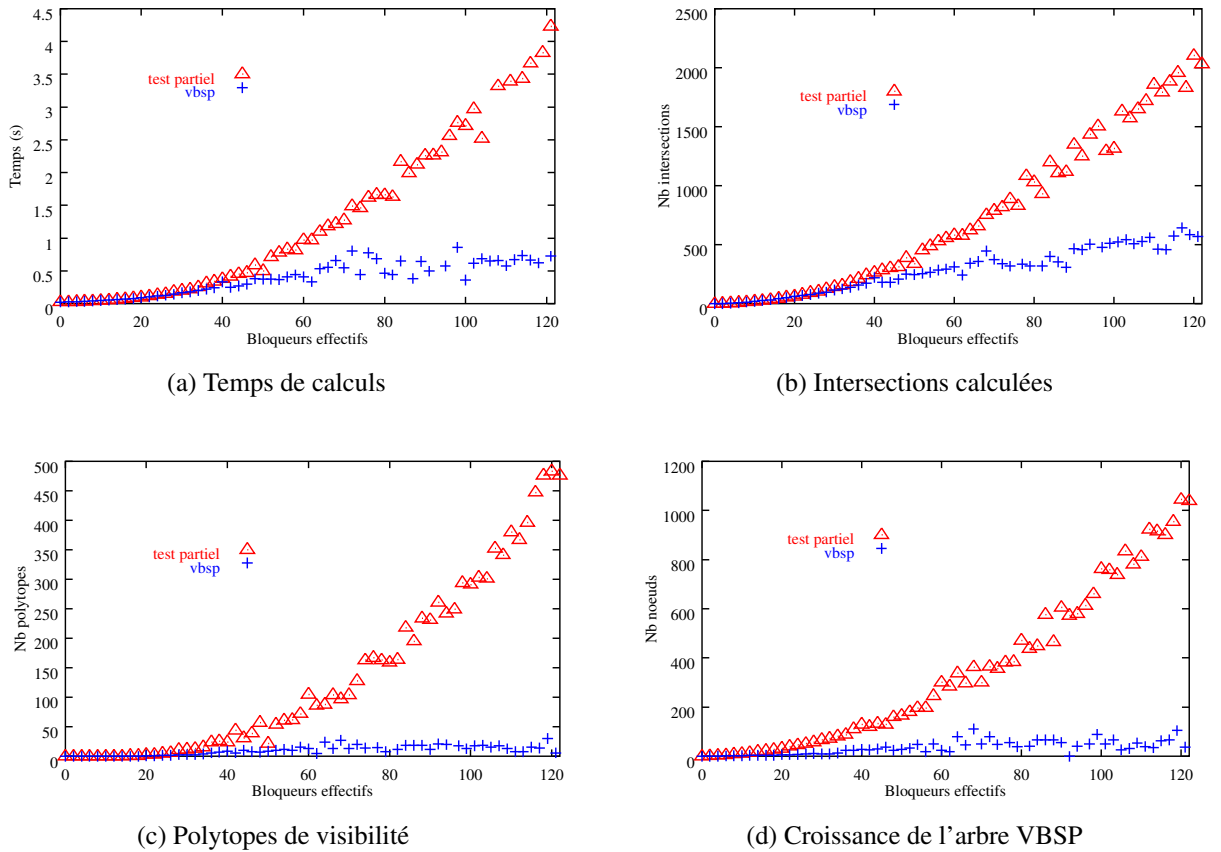


FIG. 2.14 – Résultats obtenus sur la scène extérieure (38840 polygones). L'approche proposée est bornée en temps et mémoire quand la méthode "test partiel" demeure fonction du nombre de bloqueurs effectifs.

zone des bloqueurs soustraits, et minimise le nombre d'intersections inutiles. Ceci se traduit par une réduction significative en terme de temps de calcul et d'occupation mémoire. Une autre conséquence est un comportement corrélé à la complexité visuelle plutôt qu'au nombre de bloqueurs effectifs traités.

Robustesse

Comme toute méthode analytique, le calcul de la visibilité entre deux polygones n'échappe pas aux problèmes de stabilité numérique. Et pourtant, les solutions algorithmiques à ce problème sont économes en calculs. Seules l'énumération des sommets d'un polytope et son intersection avec un hyperplan en nécessitent. Dans le premier cas, les algorithmes de Nirenstein et Haumont limitent le problème à la projection orthogonale de points sur des hyperplans dans l'espace de Plücker. Ces solutions sont clairement plus robustes qu'un algorithme général basé sur la résolution de systèmes linéaires, tel qu'utilisé par Bittner.

L'algorithme de Bajaj *et al.* [BP96] (intersection d'un hyperplan et d'un polytope) requiert la classification des sommets du polytope par rapport à un hyperplan, ainsi que l'intersection de ses arêtes par un hyperplan. Si une erreur est commise lors de la classification des sommets, l'algorithme peut ne pas converger. C'est la principale source d'instabilité que nous avons pu observer. Les erreurs commises peuvent être détectées grâce aux inconsistences engendrées sur le graphe d'incidence d'un polytope (par exemple, l'apparition d'une arête incidente à plus de deux sommets). Dans un contexte de modélisation géométrique, Paoluzzi et Pascucci [PPS04] relèvent également cette faiblesse de l'algorithme.

Notre implantation, basée sur une double précision flottante, n'échappe pas à cet écueil. A notre connaissance, il n'y a pas à ce jour d'algorithme fondamentalement différent de celui de Bajaj *et al.*. En outre celui-ci minimise déjà les opérations numériques. L'optimisation proposée par Haumont *et al.* [HMN05] n'échappe pas à l'étape de classification des sommets. Et si elle permet de se limiter au 1-squelette du polytope, elle introduit des calculs numériques supplémentaires, nécessaires à l'identification des arêtes nouvellement créées.

Une solution est l'utilisation d'une bibliothèque d'arithmétique exacte ou de précision garantie telle que la librairie CORE⁵ [Yap04b, Yap04a] afin d'assurer que le cumul des erreurs numériques commises reste inférieur à l'épsilon fixé. C'est une condition suffisante pour assurer la convergence de [BP96].

Optimisations

Notre approche pourrait encore être optimisée en terme de temps de calcul et d'occupation mémoire. Une première piste s'appuie sur la remarque suivante :

Lorsque la complexité visuelle est importante, il devient intéressant de subdiviser une requête entre deux polygones en un ensemble de sous-requêtes. Face à une complexité polynomiale, le coût de la somme des sous-requêtes peut être plus avantageux que le coût de la requête initiale. Bittner [Bit02] (voir 1.4.10) utilise ce procédé pour limiter le coût de la construction d'un arbre d'occultation. L'espace est subdivisé en régions et un arbre d'occultation est calculé pour chacune d'elles. Selon le même principe, une requête entre deux polygones peut être remplacée par une série de requêtes entre les couples d'une subdivision de ces polygones. La question qui se pose est le critère de subdivision utilisé. Bittner subdivise une région jusqu'à ce que le nombre de bloqueurs qu'elle contient soit inférieur à une valeur fixée. Ce nombre n'étant pas nécessairement corrélé à la complexité visuelle, d'autres heuristiques seraient à étudier.

Une autre optimisation possible concerne le calcul de la V-représentation d'un polytope. Nous utilisons un algorithme similaire à [Nir03]. Ce qui implique une transformation de la géométrie afin de satisfaire aux hypothèses⁶ sous-jacentes. La généralisation proposée par Haumont *et al.* [HMN05] permet d'éviter ce préalable. Nous n'avons pas effectué cette optimisation dans la mesure où la complexité de l'algorithme demeure inchangée, et sans incidence sur le coût mémoire. Elle permettrait néanmoins d'améliorer le temps d'exécution d'une requête.

⁵<http://www.cs.nyu.edu/exact/>

⁶Rappel : l'un des deux polygones est contenu dans le plan $x = 0$

2.4 Conclusion

Ce chapitre a illustré l'ensemble des opérations inutiles qui pouvaient être réalisées durant le calcul d'un complexe de polytopes au voisinage de la quadrique de Plücker. Une cause de ce problème a été identifiée : le manque d'efficacité des tests d'intersection dans les approches existantes.

En réponse, nous avons proposé l'esl-représentation d'un bloqueur, qui nous a permis de renforcer le test d'intersection d'un polytope et d'un polyèdre. Puis, nous avons présenté l'ensemble de notre approche pour calculer et représenter la visibilité de deux polygones.

Celle-ci est principalement basée sur notre test renforcé et sur une synthèse du meilleur des travaux de Nirenstein et de Bittner. Le nouvel algorithme résultant hiérarchise les tests d'intersections et localise les opérations CSG à la zone des bloqueurs soustraits. Il procure une représentation hiérarchique de la visibilité de deux polygones. Cette représentation est mémorisée dans un arbre VBSP, *i.e.* un arbre BSP dans l'espace de Plücker au voisinage de la quadrique de Plücker.

Ce chapitre montre enfin une évaluation de la complexité pratique de notre approche, aussi bien sur des configurations difficiles que sur des configurations réalistes. Il en ressort essentiellement que :

- Notre solution s'avère efficace pour pallier les problèmes identifiés en début de chapitre.
- Les complexités pratiques mesurées sur les configurations difficiles sont loin d'être aussi dissuasives que leur borne théorique.
- La complexité de notre approche s'affranchit du nombre de bloqueurs effectifs pour rester sensible à la seule complexité visuelle.

Perspectives

En dehors des possibilités d'optimisations que nous avons précédemment évoquées, d'autres pistes de travail restent ouvertes.

Les polytopes manipulés sont des ensembles contenant strictement la seule information pertinente : un sous-ensemble de droites réelles. Certes, le test d'intersection que nous avons proposé est une solution palliant les problèmes présentés en début de chapitre. Mais un autre point de vue pourrait être considéré : quel est le plus petit polytope dont l'intersection avec la quadrique de Plücker procure un ensemble donné de droites réelles ? Dans ce chapitre, nous avons également utilisé une stratégie consistant à conserver systématiquement le plus petit polytope représentatif d'un ensemble de droites réelles. Construire dès le départ un polytope minimal aiderait certainement à minimiser l'ensemble des problèmes lié à la fragmentation de la visibilité. Dans cette optique, il serait également intéressant d'évaluer dans quelle mesure les polytopes construits par les algorithmes de Nirenstein ou Haumont sont plus grands que le polytope minimal.

Une autre problématique est la mesure de la visibilité de deux polygones. Comment évaluer la "taille" de l'ensemble de droites représentées par un polytope ? Géométriquement, il s'agit de déterminer la "surface" de l'intersection d'un polytope avec la quadrique de Plücker, un calcul non trivial. Des heuristiques ont déjà été proposées. Ainsi, celle de Pu [MP96, Pu98] est à la fois basée sur la distance des esl d'un polytope, et sur leur dispersion angulaire. Bittner, quant à lui, définit une heuristique spécifique au calcul de PVS. Il l'utilise pour identifier les polytopes représentant un ensemble de visibilité négligeable ; ces derniers sont alors considérés comme dégénérés, et donc supprimés. Cependant, cette heuristique est liée au barycentre d'une face d'une cellule de vue et non à la face elle-même.

Il est difficile de statuer sur la pertinence de ces mesures. Une problématique connexe est l'échantillonnage d'un ensemble de visibilité. Haumont *et al.* choisit des visibilités dans un polytope en considérant deux points par combinaisons linéaires de ses sommets. L'intersection du segment formé avec la qua-

drique de Plücker procure une droite réelle contenue dans le polytope. En revanche, l'échantillonnage de l'ensemble des visibilitées relativement à un taux d'erreur fixé dépend de la taille de cet ensemble. D'où la nécessité de savoir en faire une évaluation fiable. Dans l'optique de maîtriser et exploiter au mieux l'information calculée, ces problèmes méritent à terme d'être traités. Toutefois, comme nous allons le voir dans le chapitre suivant, la seule information contenue dans un arbre VBSP permet déjà de proposer de nouvelles techniques.

Ombres douces en synthèse d'images

Sommaire

| | | |
|------------|--|-----------|
| 3.1 | Calcul d'ombres douces | 62 |
| 3.1.1 | Subdivision de surfaces | 62 |
| 3.1.2 | Volumes d'ombre | 62 |
| 3.1.3 | Approches stochastiques | 63 |
| 3.1.4 | Faisceaux de rayons | 64 |
| 3.1.5 | Volumes d'ombre et de pénombre | 65 |
| 3.1.6 | Silhouette étendue et volume d'ombre douce | 68 |
| 3.1.7 | Maillage de discontinuités et événements visuels | 69 |
| 3.2 | Approche proposée | 70 |
| 3.2.1 | Vue d'ensemble | 70 |
| 3.2.2 | Visibilité point-polygone | 71 |
| 3.2.3 | Résultats | 74 |
| 3.3 | Discussion et perspectives | 81 |
| 3.3.1 | Extensions et optimisations | 81 |
| 3.3.2 | Limitations actuelles | 81 |
| 3.3.3 | Perspectives | 82 |

Ce chapitre présente une nouvelle approche pour le rendu avec ombres douces via un calcul analytique de l'illumination directe. Celui-ci repose d'une part sur le précalcul de la visibilité des sources lumineuses, d'autre part sur un algorithme permettant d'extraire efficacement la visibilité d'une source depuis un point. L'illumination directe peut alors être calculée analytiquement, garantissant une qualité d'image optimale.

Après un rappel des principales méthodes existantes, nous présentons notre solution, en insistant plus particulièrement sur notre algorithme d'extraction de la visibilité depuis un point. Les résultats discutés en fin de chapitre indiquent que le calcul analytique de l'illumination directe peut être porté à un niveau de performance supérieur ou égal à celui des méthodes stochastiques les plus répandues à ce jour.

3.1 Calcul d'ombres douces

Cette section présente un état de l'art des principales techniques utilisées pour le rendu des ombres douces. Quelques méthodes relatives aux calculs des ombres dures sont également abordées dans la mesure où elles ont été sources d'inspiration pour le calcul des ombres douces. Considérant notre contexte, nous nous intéressons essentiellement aux méthodes analytiques. Les méthodes matérielles, exploitant le processeur des cartes graphiques (GPU), ne seront pas détaillées. Le lecteur intéressé peut néanmoins se référer à l'état de l'art proposé par Hasenfratz *et al.* [HLHS03].

3.1.1 Subdivision de surfaces

Weiler et Atherton [WA77] ont proposé un algorithme déterminant exactement les parties visibles d'un polygone depuis un point de vue donné. Les polygones sont traités par ordre de profondeur depuis le point de vue. Les plus proches sont utilisés pour intersecter dans le plan image les plus éloignés. Ceux totalement masqués sont alors progressivement supprimés. Cet algorithme peut être utilisé en deux passes pour le calcul d'ombres dures [AWG78]. Durant la première passe, la caméra est confondue avec la source ponctuelle. Les polygones sont alors subdivisés en fragments éclairés ou dans l'ombre. La seconde passe est effectuée depuis le point de vue en tenant compte du marquage préalablement établi.

Cette approche à deux passes a également été utilisée par Nishita *et al.* [NN74] sur la base d'un algorithme exact 2D de suppression des arêtes masquées [NN72].

3.1.2 Volumes d'ombre

Crow [Cro77] définit le volume d'ombre d'un objet vis-à-vis d'une source ponctuelle. Considérant le maillage polygonal de cet objet, l'algorithme identifie les arêtes silhouettes (*i.e.* les arêtes incidentes à deux polygones, l'un orienté vers la source, l'autre non). Avec la source, les arêtes silhouettes forment des plans qui délimitent le volume d'ombre de l'objet. Seuls les points de l'espace non contenus dans un tel volume sont éclairés par la source (voir figure 3.1.(a)). Pendant la phase de rendu, un compteur est incrémenté ou décrémenté lorsqu'un rayon entre ou sort d'un volume d'ombre. Un compteur nul au point d'impact signifie que ce dernier est visible depuis la source. Ce mécanisme est illustré par la figure 3.1.(b). Bergeron [Ber86] généralise cette approche à des polygones concaves ou non plans et prend en compte les maillages non fermés.

Chin et Feiner [CF89] utilisent un arbre BSP pour représenter les volumes d'ombres. Les nœuds internes contiennent les plans délimitant les volumes d'ombres, et chaque feuille représente une région soit dans l'ombre soit en visibilité de la source ponctuelle. Pour cela, les polygones sont traités par ordre croissant d'éloignement depuis la source ponctuelle. Ils sont insérés dans l'arbre BSP et intersectés par les plans délimitant les volumes d'ombres déjà insérés. L'algorithme détermine ainsi une subdivision polygonale où chaque élément est soit dans un volume d'ombre, soit éclairé. Dans ce dernier cas, le volume d'ombre associé est ajouté à l'arbre. Cette approche a été reprise par Chrysanthou *et al.* [CS95] pour le calcul d'ombres dures dans des scènes dynamiques. Elle a aussi été modifiée pour le calcul hiérarchique de la visibilité depuis un point d'observation [BHS98].

Méthode populaire, les volumes d'ombre ont fait l'objet d'implantations [Hei91] et d'optimisations [EK03, LWGM04] matérielles. Nous ne détaillons pas ces variantes ici, car trop éloignées de la problématique analytique qui est la nôtre.

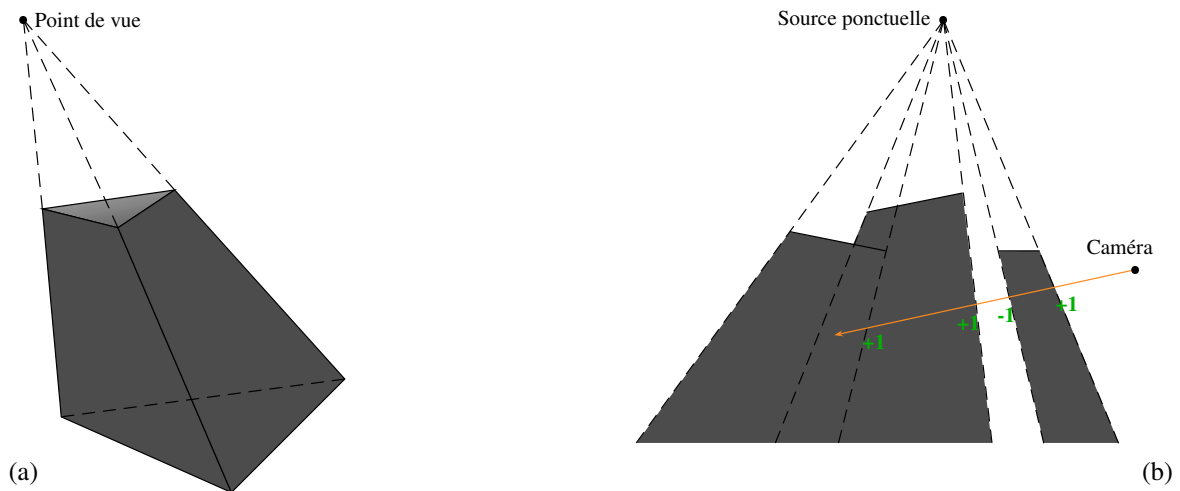


FIG. 3.1 – (a) le volume d'ombre engendré par un triangle depuis un point de vue. (b) le point d'impact d'un rayon issu d'un point de vue n'est pas visible depuis la source lumineuse ponctuelle si le rayon est sorti moins de fois qu'il n'est entré dans un volume d'ombre.

3.1.3 Approches stochastiques

Les approches stochastiques [CPC84, Co086] consistent à échantillonner une source lumineuse (non ponctuelle) selon une approche de type Monte-Carlo. La contribution d'une source en un point est approchée en lançant un rayon (ou *shadow ray* [Whi80]) vers chaque échantillon de la source, et en sommant les contributions de chaque échantillon visible. La figure 3.2 illustre le procédé. La problématique d'une source surfacique ou volumique est ainsi ramenée à celle d'un ensemble de sources ponctuelles. Mais il est souvent nécessaire d'utiliser un grand nombre d'échantillons pour converger vers une solution crédible, qui demeure malgré tout une approximation. Pour une loi de distribution uniforme, diviser par deux l'erreur commise implique de multiplier par quatre le nombre d'échantillons choisis. L'évaluation de l'illumination en un point devient ainsi très coûteuse, particulièrement lorsque la superficie des sources est importante.

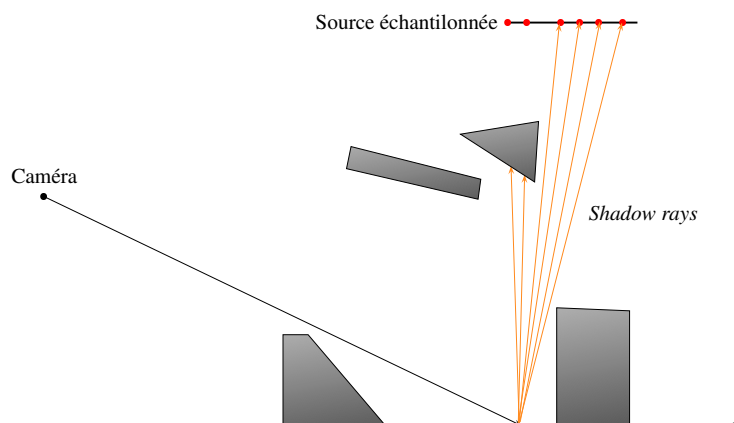


FIG. 3.2 – Illustration 2D d'une approche stochastique

Deux méthodes sont alors principalement utilisées pour réduire l'erreur commise. La première consiste à utiliser un échantillonnage stratifié : la source est subdivisée régulièrement et chaque partie est ensuite individuellement échantillonnée uniformément, en général en considérant un unique échantillon. La seconde porte le nom d'échantillonnage d'importance [SWZ96, SSSK04] : il s'agit alors d'exploiter la connaissance que l'on peut avoir de la fonction de visibilité afin de privilégier l'échantillonnage de certaines parties de la source. Même si ces méthodes permettent de converger plus rapidement vers une solution acceptable, elles nécessitent toujours de lancer un grand nombre de rayons. De fait, elles sont toujours accompagnées d'un lancer de rayons optimisé.

Le *light buffer* [HG86] est une structure accélératrice spécifique aux sources lumineuses. Elle mémorise pour chaque échantillon d'une source surfacique la liste des bloqueurs potentiels en fonction de l'angle d'incidence des rayons. Les bloqueurs à tester pour un *shadow ray* sont ainsi rapidement déterminés. Concernant le lancer de rayons en général, de nombreuses optimisations ont été développées. Elles reposent essentiellement sur des subdivisions hiérarchiques de l'espace, afin d'accélérer les tests d'intersection d'un triangle par un rayon. Le lecteur intéressé trouvera davantage de détails, ainsi qu'une étude comparative de ces différentes structures (grille régulière, octree, kd-tree, hiérarchie de boîtes englobantes...), dans les travaux de Vlastimil Havran [HPP00, Hav00, HP03]. En matière d'efficacité, les travaux d'Ingo Wald [WBWS01, Wal04] font également référence en matière de lancer de rayons interactif. Ils sont aujourd'hui à l'origine de la librairie commerciale OpenRT¹ [WBS02, DWBS03] qui se veut une alternative future aux cartes graphiques actuelles. Dans cet optique, de nouvelles cartes² sont à l'étude, entièrement basées lancer de rayons [SWS05].

3.1.4 Faisceaux de rayons

Les méthodes basées sur les faisceaux de rayons reposent sur l'observation suivante : des rayons voisins ont une forte probabilité de suivre des trajets similaires. Pour exploiter cette cohérence spatiale, des faisceaux de rayons sont utilisés. Amanatides [Ama84] utilise des cônes pour représenter un faisceau. Les ombres douces sont supportées en lançant un cône depuis un point de réception vers la source lumineuse (sphérique ou circulaire de préférence). Cependant la visibilité de la source n'est qu'approchée car l'occultation entre bloqueurs n'est pas prise en compte.

Heckbert *et al.* [HH84] utilise une approche différente dite de "tracés de faisceaux" (ou *beam tracing*). La caméra et la fenêtre de vue définissent une pyramide lancée à travers la scène. La géométrie de la pyramide est récursivement intersectée avec la géométrie de la scène. La méthode supporte les réflexions et approche les réfractions. Comme pour le lancer de cônes, une pyramide lancée vers une source surfacique permet le calcul des ombres douces. Mais, ici, l'occultation entre bloqueurs est prise en compte. Un problème de cette approche est que les interactions et intersections successives peuvent donner une forme très complexe aux faisceaux.

Shinya *et al.* [STN87] formalise le principe du lancer de faisceaux en proposant le *pencil tracing*. Dans ce cas, un faisceau est formé d'un rayon auquel sont associées les déviations possibles de son axe. Cette formulation supporte mieux la réfraction et permet de garantir une qualité d'image par estimation d'un taux d'erreur. A notre connaissance, le support de sources non ponctuelles n'a pas été proposé.

Ghazanfarpour *et al.* [GH98] proposent une solution similaire à [HH84] mais qui ne nécessite aucune intersection du faisceau avec la géométrie de la scène. Les faisceaux sont lancés à travers chaque pixel de l'image et récursivement subdivisés jusqu'à ce qu'ils n'intersectent qu'un même polygone ou bien

¹<http://www.openrt.de>

²<http://www.saarcor.de>

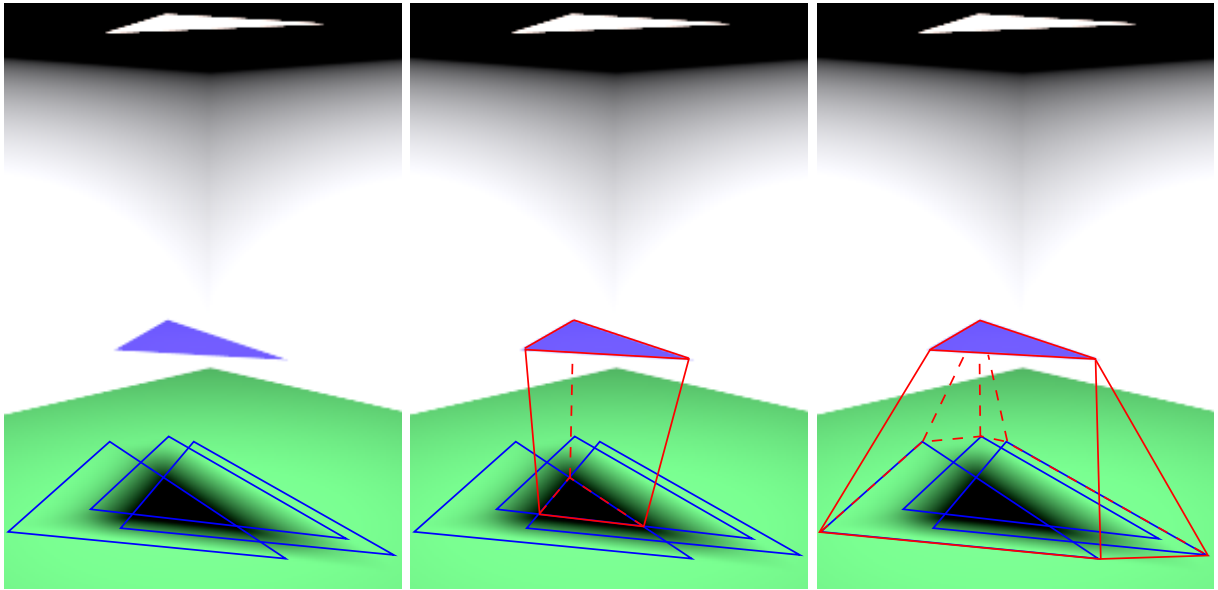


FIG. 3.3 – Volume d'ombre et de pénombre. A gauche : l'empreinte au sol des volumes d'ombre induits par les sommets de la source lumineuse. Au centre : le volume d'ombre induit par la source est l'intersection des volumes d'ombre induits par chacun de ses sommets. A droite : le volume de pénombre induit par la source est l'enveloppe convexe des volumes d'ombre induit par chacun de ses sommets (valable seulement si le plan support au bloqueur n'intersecte pas la source).

jusqu'à ce qu'une profondeur maximale de subdivision soit atteinte. Cette approche supporte les sources surfaciques. Un faisceau comprenant la source est lancé en direction d'une surface et récursivement subdivisé jusqu'à en identifier les parties visibles ou masquées. Si la profondeur maximale est atteinte, l'élément de surface atteint par le sous-faisceau est classé partiellement visible. Dans ce cas des approximations sont possibles pour le calcul de l'illumination.

3.1.5 Volumes d'ombre et de pénombre

Nishita *et al.* [NN85, NON85] étendent la notion de volume d'ombre introduite par Crow [Cro77] à des sources polygonales ou polyédriques convexes. Considérant les volumes d'ombre associés à chaque sommet de la source, l'ombre et la pénombre d'un objet convexe sont respectivement l'intersection des volumes et l'enveloppe convexe de leur union³. La figure 3.3 en donne l'illustration.

L'algorithme de Crow est utilisé pour construire les volumes d'ombre et de pénombre induits par une source surfacique. Ils sont intersectés avec la scène afin d'obtenir une subdivision où chaque polygone est classé soit éclairé, soit dans la pénombre, soit dans l'ombre. L'illumination d'un point éclairé est calculée analytiquement par intégration le long des contours de la source [NN83]. Pour un point P dans la pénombre, le calcul se fait en deux temps. D'abord, les parties visibles P_v^i de la source depuis P sont déterminées [NN74]. Puis l'illumination est calculée pour chacun des fragments visibles P_v^i et leurs contributions en P sont additionnées. Pour que l'approche demeure praticable, ces calculs sont uniquement réalisés aux sommets d'un maillage régulier de la scène. Les contributions aux autres points sont obtenues par interpolations.

³Cette affirmation ne vaut que si le plan support au bloqueur n'intersecte pas la source. Dans cette dernière configuration, Teller [Tel92a], dans sa thèse, donne un exemple d'antipénombre disconnectée

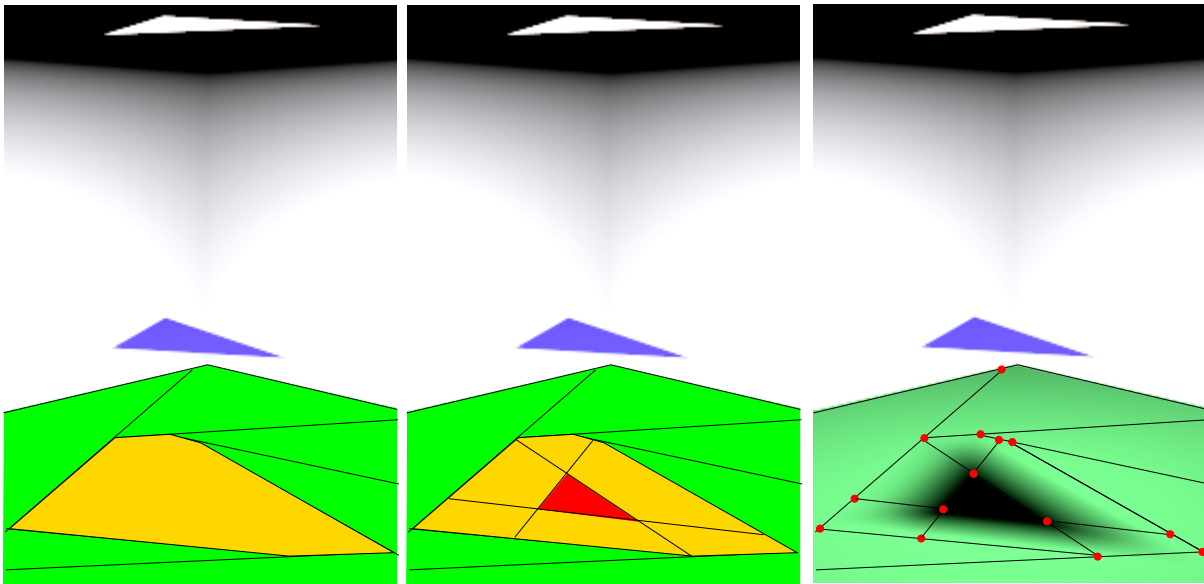


FIG. 3.4 – Illustration de la méthode de Chin et Feiner [CF92]. A gauche : les polygones de la scène sont “filtrés” à travers la représentation BSP des volumes de pénombre. Il en résulte une subdivision dont chaque élément est classé visible ou dans la pénombre. Au centre : une seconde passe à travers la représentation BSP des volumes d’ombre subdivise les éléments classés dans la pénombre et isole les parties totalement contenues dans l’ombre. A droite : l’illumination est calculée analytiquement aux sommets des éléments visibles ou dans la pénombre. Les valeurs aux autres points sont calculées par interpolation.

Dans le même esprit, Chin et Feiner [CF92] étendent leur algorithme pour le calcul d’ombres dures [CF89] à des sources polygonales. Deux arbres BSP sont construits pour représenter respectivement les volumes de pénombre et les volumes d’ombre. L’algorithme traite les polygones par ordre de profondeur depuis la source. Chaque polygone est inséré dans l’arbre BSP de pénombres afin d’en déterminer les parties visibles ou partiellement visibles. Seuls les fragments contenus dans une zone de pénombre sont ensuite insérés dans l’arbre BSP d’ombres afin d’en éliminer les parties masquées de la source. Il en résulte une subdivision polygonale convexe de la scène dont chaque élément est classé visible, partiellement visible, ou invisible. La figure 3.4 résume le procédé. L’illumination est alors calculée similairement à [NN85].

L’avantage d’une représentation BSP des volumes d’ombre et pénombre est la fusion partielle de l’information de visibilité depuis la source. Cependant, cette fusion n’est que partielle. En effet, les volumes d’ombre et de pénombre engendrés par la conjonction de plusieurs bloqueurs ne sont pas descriptibles uniquement par des plans : des événements EEE associés à des surfaces réglées apparaissent. Un arbre BSP dans \mathbb{R}^3 ne permet pas la représentation de ces surfaces. L’information de visibilité s’en trouve faussée et des régions dans l’ombre peuvent être classées par erreur dans la pénombre (ce problème est également présent dans [NN85]). Ces erreurs de classification sont “récupérées” par le calcul exact de la visibilité de la source aux sommets des fragments considérés dans la pénombre.

Tanaka *et al.* [TT97] considèrent une subdivision régulière des rayons issus d’une source polygonale. Celle-ci guide une segmentation de l’espace 3D en régions auxquelles est associée la liste des bloqueurs contenus. Pendant la phase de rendu, la structure procure une liste conservatrice de bloqueurs susceptibles d’affecter la visibilité de la source depuis un point. Cette liste est affinée en éliminant tous ceux dont le volume englobant n’intersectent pas la pyramide définie par la source et le point. La silhouette de chaque

bloqueur restant est projetée sur la source afin d'en supprimer les parties masquées et l'illumination calculée analytiquement.

A la différence des deux approches précédentes [NN85, CF92], le procédé permet de réduire la géométrie à projeter sur la source et donc de minimiser le coût lié au calcul de ses parties visibles. En outre, l'algorithme est appliqué en chaque point visible depuis la caméra, et pas seulement aux sommets d'une subdivision.

Les trois méthodes présentées ci-dessus ont en commun un calcul analytique de l'illumination directe. Un tel calcul est précis et garantit le réalisme du rendu. Toutefois, il nécessite de déterminer en un point considéré quelles sont les parties visibles d'une source. De plus, cela n'est possible que pour des sources d'émission uniforme [NN83] ou jacobienne [Arv94].

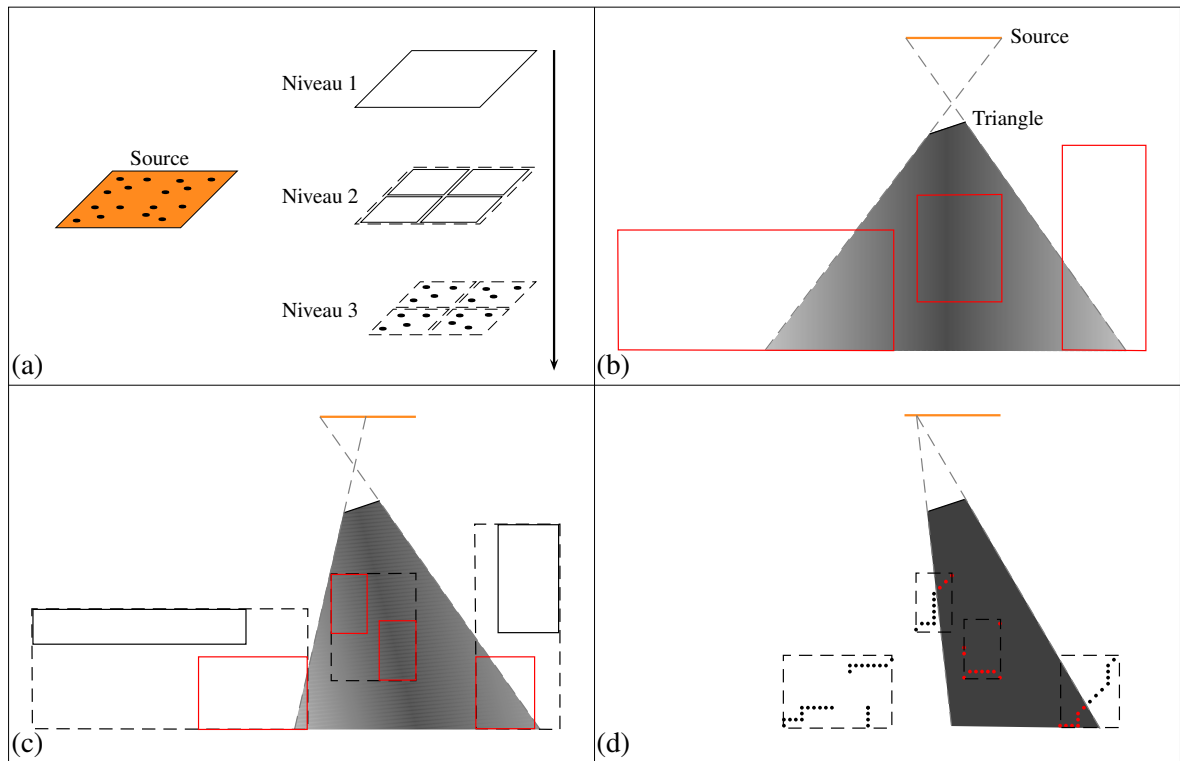


FIG. 3.5 – Illustration de [LA05]. (a) La source échantillonnée est représentée hiérarchiquement sur 3 niveaux. Le troisième niveau contient les points d'échantillonnage de la source. (b) Les récepteurs (*i.e.* les points en visibilité de la caméra obtenus par z-buffer) sont organisés dans une hiérarchie de boîtes englobantes. Le volume de pénombre d'un triangle depuis le niveau 1 de la source est intersecté avec la hiérarchie spatiale afin d'identifier les régions pour lesquelles le triangle masque potentiellement la source. (c) Cette sélection est affinée en répétant l'opération avec le niveau 2 de la source. (d) Lorsqu'une feuille est atteinte, l'algorithme teste si ses récepteurs sont contenus dans les volumes d'ombre du triangle et des échantillons du niveau 3 de la source.

Laine *et al.* [LA05] proposent une approche basée sur la traversée simultanée de deux structures hiérarchiques. Leur méthode, analytique, s'inspire de deux algorithmes matériels [AAM04, AL04]. La scène est projetée (via un z-buffer) dans le plan image depuis la position de la caméra afin d'obtenir l'ensemble des points visibles, qui sont de fait les récepteurs. Ces derniers sont organisés dans une hiérarchie de boîtes englobantes. Par ailleurs, la source est échantillonnée et représentée sur 3 niveaux (figure 3.5.a). Le principe de l'algorithme est alors de déterminer pour chaque triangle les couples (récepteur, échantillon) qu'il masque. La figure 3.5 illustre le procédé. Chaque récepteur mémorise les échantillons qui

lui sont masqués par un triangle. Lorsque tous les triangles sont traités, l'illumination en chaque récepteur est calculée uniquement pour les échantillons visibles restants. Une implantation optimisée porte le procédé à un haut niveau de performance. Cependant, les récepteurs devant être connus par avance, la méthode est dépendante de la position de la caméra. Pour la même raison, un anti-aliasage adaptatif ne peut être pris en compte simultanément.

3.1.6 Silhouette étendue et volume d'ombre douce

Akenine-Möller *et al.* [AMA02] étendent à une surface la notion d'arête silhouette pour un point de vue. Le but recherché est de travailler avec la silhouette (étendue) des objets, en général beaucoup moins complexe que l'ensemble des polygones qui les constituent [McG04]. À l'image des volumes d'ombre ou de pénombre, le volume de pénombre d'une arête silhouette (*penumbra wedge*) définit la région de l'espace dont elle affecte la visibilité de la source. À la différence d'un volume de pénombre d'un polygone, qui contient son volume d'ombre, la région décrite est en visibilité au moins partielle de la source. On parle aussi de "volume d'ombre douce". La figure 3.6 illustre l'identification d'une arête silhouette pour une source surfacique ainsi que le volume de pénombre qu'elle engendre.

Sur la base de cette idée, Assarsson *et al.* [ADMAM03, AAM03] développent un algorithme matériel pour le rendu interactif (50 fps et plus) d'ombres douces convaincantes (en raison d'hypothèses simplificatrices).

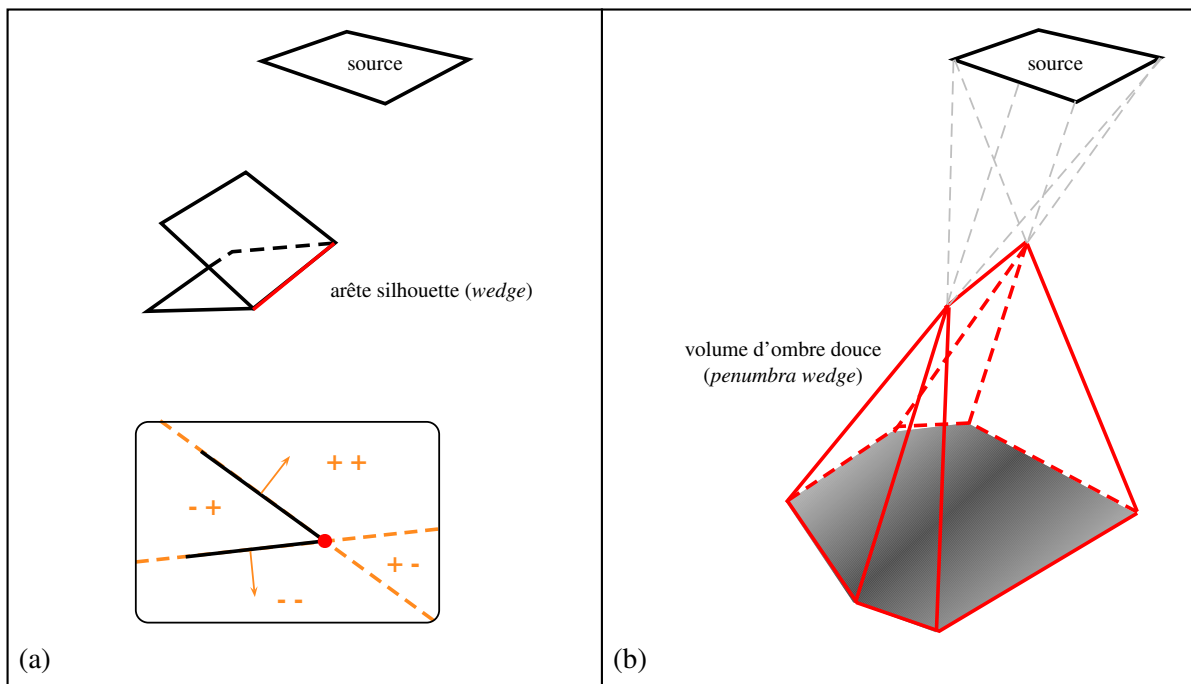


FIG. 3.6 – (a) Extension à une source surfacique de la notion d'arête silhouette. Considérons deux polygones adjacents par une arête. Les plans supports aux polygones partitionnent l'espace 3D en 4 régions (coupe 2D en bas). L'arête est silhouette si la source n'est pas entièrement contenue dans les espaces (+ -) ou (- +). (b) Volume de pénombre ou volume d'ombre douce induit par une arête silhouette vis-à-vis d'une source. (*illustrations adaptées de [LAA⁺05]*)

Laine *et al.* [LAA⁺05] développent une approche analytique reprenant les principes présentés dans [ADMAM03, AAM03]. L'algorithme utilise un hémicube contenant la scène. Pour chaque arête silhouette (étendue) identifiée, son volume de pénombre est projeté sur l'hémicube. Son empreinte est mé-

morisée conservativement et hiérarchiquement grâce à des grilles multi-résolutions. Pendant la phase de rendu, chaque point de réception P est projeté depuis la source sur l'hémicube. Un ensemble conservatif d'arêtes silhouettes pour P est obtenu. Les arêtes silhouettes valides sont projetées sur la source depuis P et permettent de reconstruire la fonction de visibilité. Un mécanisme de compteurs de profondeur permet de déterminer quels points d'échantillonnage sont visibles depuis P en lançant un unique rayon (*shadow ray*) vers la source.

3.1.7 Maillage de discontinuités et événements visuels

Un maillage de discontinuités [Hec92] subdivise la géométrie de la scène en essayant de prendre en compte les événements visuels. Le principe est d'obtenir des mailles pour lesquelles la fonction de visibilité peut être considérée constante. L'application de cette méthode concerne davantage les calculs d'illumination globale. Le maillage de discontinuité échoue à traiter des scènes de grande taille en raison des ressources mémoires qu'il nécessite.

Drettakis *et al.* [DF94] se servent de projections arrières pour améliorer la construction d'un maillage de discontinuités. Stewart *et al.* [SG94] ont également exploré cette approche.

Durand *et al.* [DDP98, DDP99] utilisent le squelette de visibilité [DDP97] (voir aussi 1.4.7) pour guider la construction d'un maillage de discontinuités. Le squelette permet également d'extraire exactement la visibilité d'une source pour le calcul du facteur de forme. Cependant, la structure ne contenant pas toute l'information de visibilité, seules les droites possédant au plus un degré de liberté sont représentées. Par conséquent, ce calcul n'est possible qu'aux sommets existants de la scène.

La version robuste du squelette [DD02] est aussi appliquée au rendu d'ombres dures et douces, mais pour que le calcul demeure efficace, tous les événements visuels ne sont pas considérés.

3.2 Approche proposée

Cette section présente un algorithme pour le calcul d'ombres douces en illumination directe. Celui-ci repose sur le précalcul de la visibilité des sources avec les polygones de la scène. Après avoir donné une vue d'ensemble de notre approche, nous montrons comment la visibilité depuis un point peut être efficacement extraite d'un arbre VBSP. Cet algorithme d'extraction constitue la clef de notre approche dont nous évaluons et discutons les résultats à la fin de ce chapitre.

3.2.1 Vue d'ensemble

Nous proposons une nouvelle approche pour un rendu haute qualité des ombres douces. Nous nous plaçons dans des environnements composés de polygones convexes. Les sources lumineuses auxquelles nous faisons référence sont toutes surfaciques, et, pour que le calcul analytique de l'illumination directe soit possible, elles sont également d'émission uniforme. Pour chaque polygone de la scène, nous précalculons sa visibilité avec chaque source lumineuse. Un arbre VBSP (voir le précédent chapitre) encode cette information pour chaque couple (polygone, source). Ce précalcul ne dépendant pas de la position de la caméra, il est effectué une seule fois et mémorisé dans un fichier.

Le rendu d'une scène se déroule comme suit :

- Pour chaque rayon primaire rencontrant un polygone B en un point w
 - Pour chaque source lumineuse A visible de B
 - ▷ Prendre l'arbre VBSP associé à A et B .
 - ▷ Extraire de l'arbre une subdivision convexe des parties visibles de A depuis w .
 - ▷ Pour chaque partie visible
 - ◇ Calculer analytiquement sa contribution en w
 - ◇ Sommer le résultat à l'énergie totale reçue en w

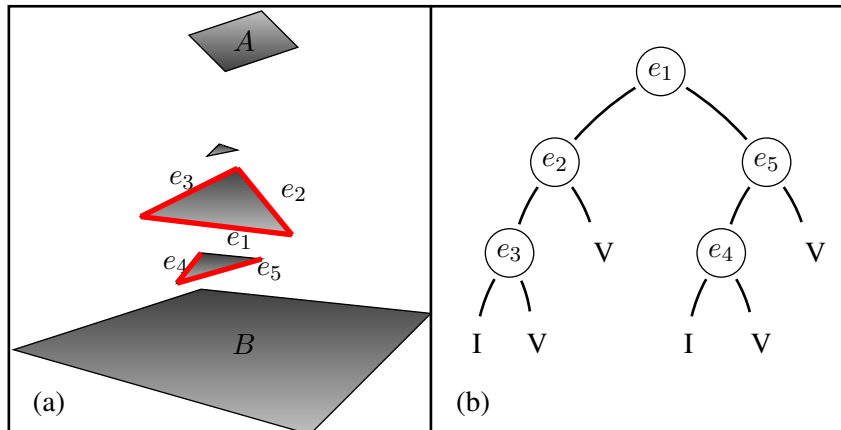


FIG. 3.7 – Le calcul d'un arbre VBSP prend en compte la fusion des bloqueurs. Cette fusion est effective depuis n'importe quel point du polygone A ou du polygone B .

Cette approche se distingue des précédents travaux sur plusieurs points. Tout d'abord, le précalcul de la visibilité unifie volumes d'ombre et pénombre au sein d'une même structure. De plus, ce calcul effectuant la fusion des bloqueurs dans l'espace de Plücker, l'occultation est totalement prise en compte (3.7). A notre connaissance, ce n'est le cas pour aucune autre approche existante à ce jour. Elle se démarque aussi

des méthodes basées sur une subdivision de la scène, où un calcul précis de l'illumination est effectué en ses sommets. L'approche proposée ne nécessite aucun maillage ni subdivision, et l'illumination directe est obtenue avec un même niveau de précision en n'importe quel point de la scène.

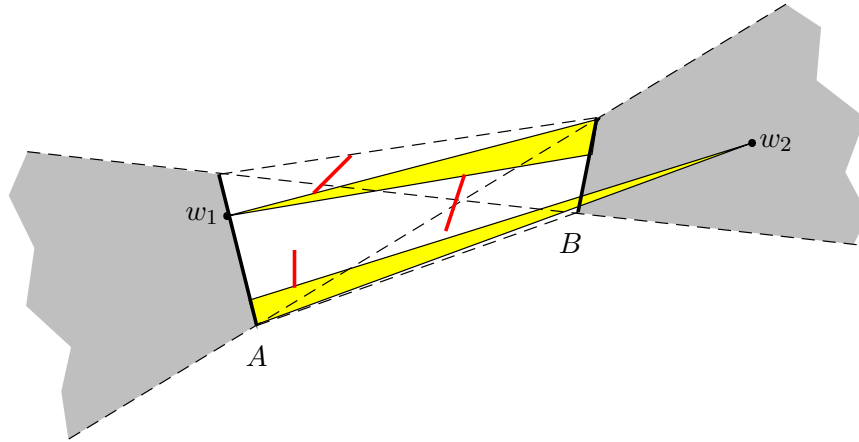


FIG. 3.8 – La visibilité de B depuis w_1 est associable à l'ensemble des droites intersectant uniquement w_1 et B . C'est un sous-ensemble de la visibilité globale de A et B . Ceci s'étend à n'importe quel point situé dans les zones délimitées à gauche de A et à droite de B . Ainsi l'ensemble de droites n'intersectant que A , B et w_2 définit la visibilité de A depuis w_2 à travers B .

La clef de notre méthode consiste à extraire d'un arbre VBSP la visibilité depuis un point. Comme l'illustre la figure 3.8, l'idée est d'exploiter le fait que la visibilité depuis un point sur un polygone est un sous-ensemble de la visibilité depuis ce polygone. La section suivante présente l'algorithme qui permet d'extraire cette information sous une forme directement manipulable dans \mathbb{R}^3 .

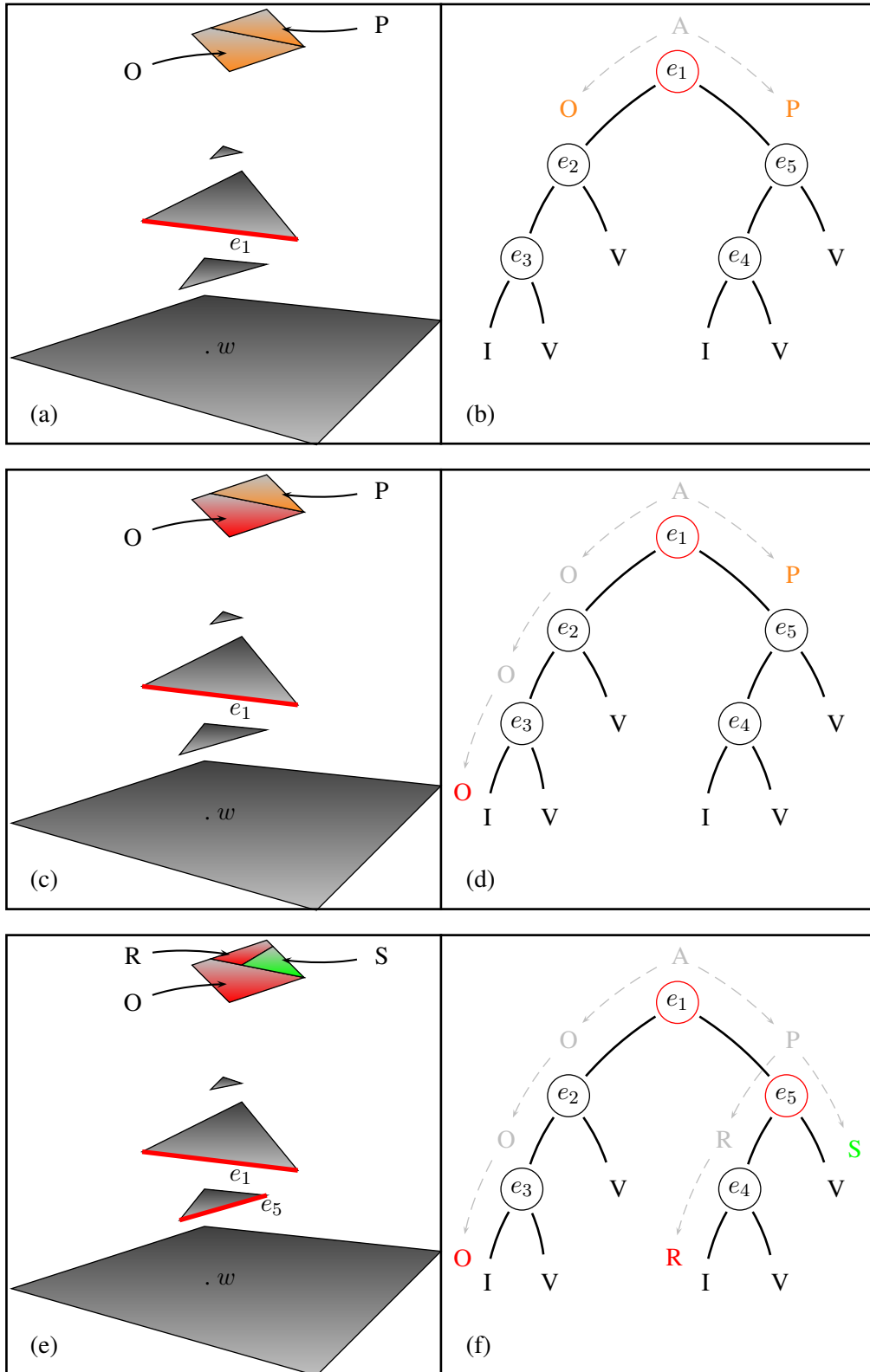
3.2.2 Visibilité point-polygone

Considérons deux polygones convexes A et B et l'arbre VBSP encodant leur visibilité dans l'espace de Plücker. Étant donné un point w sur le polygone A (resp. B), nous souhaitons extraire la visibilité de B (resp. A) depuis le point w .

Commençons par formuler ce problème dans l'espace de Plücker. Soient deux droites distinctes l_1 et l_2 s'intersectant en w . Dans l'espace de Plücker, toute droite passant par w est incidente aux hyperplans $h_{l_1^*}$ et $h_{l_2^*}$ duaux de l_1 et l_2 . Le sous-ensemble des visibilités passant par w est l'intersection par $h_{l_1^*} \cap h_{l_2^*}$ de la partition décrite par l'arbre VBSP. Pratiquement, il s'agit de calculer l'intersection des deux hyperplans et de l'arbre VBSP, afin d'obtenir un ensemble de 3-polytopes. Leur intersection avec la quadrique de Plücker est l'ensemble 2D des visibilités de A et B passant par le point w .

Cette approche soulève deux problèmes. Tout d'abord, elle nécessite des calculs non triviaux et coûteux. Ensuite il est difficile d'en exploiter le résultat. En effet, celui-ci n'est pas directement manipulable dans \mathbb{R}^3 . On obtient l'ensemble des droites passant par un point et un polygone, et non une subdivision des parties visibles de ce polygone. Certes, ces deux représentations sont équivalentes, et l'on pourrait passer de l'une à l'autre. Malheureusement le coût d'une telle méthode deviendrait prohibitif.

Nous proposons une solution alternative. La visibilité depuis un point de \mathbb{R}^3 est un problème 3D par nature. Un problème se résolvant généralement plus efficacement dans l'espace où il s'exprime le mieux, nous proposons un algorithme d'extraction opérant entièrement dans \mathbb{R}^3 . Aussi, pour tout nœud d'un arbre VBSP, on ne considère plus un hyperplan, mais la droite orientée qui lui correspond dans \mathbb{R}^3 . Par construction, chaque droite est support à une arête d'un polygone de la scène. Ce sont ces arêtes que nous

FIG. 3.9 – Extraction de la visibilité du polygone A depuis un point w sur le polygone B

Algorithme 3 : Pseudo-code de l'algorithme d'extraction de la visibilité d'un polygone depuis un point. La fonction `VisibiliteP2P` fait référence au précédent chapitre et retourne l'arbre VBSP associé à deux polygones

```

1 Polygone  $B, A$ 
2 VBSP  $H \leftarrow \text{VisibiliteP2P}(B, A)$ 
3 Point  $w \leftarrow$  un point sur  $B$ 
4 Polygone2D  $Poly \leftarrow \text{Copie2D}(A)$ 
5 Polygones  $V\text{Fragments} \leftarrow \emptyset$ 

6 procedure FPQuery (VBSP  $H$  ; Polygone2D  $Poly$ )
7   Plan3D  $plan$ 
8   begin
9     if feuilleVisible ( $H$ ) then
10      |  $V\text{Fragments} \leftarrow V\text{Fragments} \cup Poly$ 
11    else
12      if noeudInterne ( $H$ ) then
13        |  $droite \leftarrow \text{projection}(w, H.\text{areteOrientee})$ 
14        | switch  $Poly \cap droite$  do
15          | case POSITIF
16          |   | FPQuery ( $H.\text{pos}, Poly$ )
17          | case NEGATIF
18          |   | FPQuery ( $H.\text{neg}, Poly$ )
19          | case INTERSECTE
20          |   | FPQuery ( $H.\text{pos}, Poly^+$ )
21          |   | FPQuery ( $H.\text{neg}, Poly^-$ )
22          | end
23        | end
24      | end
25    end
26 end

```

stockons dans les nœuds d'un arbre VBSP.

Supposons que w soit sur B . Le polygone A est représentatif de toutes les droites passant par w et intersectant A . Le principe de l'algorithme d'extraction est de "filtrer" A à travers l'arbre VBSP de A et B . Pour chaque nœud rencontré contenant une arête e_i , on considère l'événement visuel VE généré par w et e_i . La surface associée à cet événement sert à subdiviser le polygone A afin d'en extraire les parties convexes visibles et invisibles.

La figure 3.9 propose un exemple d'extraction des parties visibles d'un polygone A depuis un point w sur un polygone B . L'algorithme débute à la racine de l'arbre VBSP associé à A et B . La position de A est testée par rapport au plan défini par l'arête orientée e_1 et w (3.9(a & b)). L'intersection de ce plan avec A produit deux fragments O et P . L'algorithme se poursuit dans les sous-arbres gauche et droit avec les fragments correspondants (3.9(c & d)). Ici, le fragment O atteint une feuille marquée invisible. Cette partie de A n'est donc pas visible depuis w (3.9(e & f)). Le fragment P est encore une fois intersecté, donnant un fragment R invisible et un fragment S visible. Au final l'algorithme procure une subdivision convexe des parties de A visibles ou invisibles depuis w .

Plutôt que d'effectuer des calculs d'intersections plan/polygone en 3D, une optimisation consiste à projeter depuis w la droite support d'une arête e_i dans le plan du polygone A . On se ramène ainsi à des intersections 2D demi-espace/polygone.

Le pseudo-code d'une requête d'extraction est donné par l'algorithme 3. Une requête débute à la racine de l'arbre VBSP (ligne 2) avec une copie du polygone A projetée sur son plan support (ligne 4). S'il s'agit d'une feuille *visible* (ligne 9), le fragment est ajouté à l'ensemble des parties visibles de A depuis w (ligne 10). Si c'est un noeud interne, alors l'arête orientée e_i contenue dans le noeud courant est projetée depuis w sur le plan support de A , formant une droite 2D dont l'orientation est consistante avec e_i (ligne 13). Puis la position du fragment est testée. S'il est contenu dans le demi-espace positif ou négatif induit par la droite, l'algorithme se poursuit récursivement dans le sous-arbre approprié (lignes 15-18). En cas d'intersection, il se poursuit dans les deux sous-arbres avec les fragments produits de cette intersection (ligne 19-21). L'algorithme produit une subdivision polygonale convexe de A dont chaque élément est soit totalement visible, soit totalement invisible depuis w . Seuls les éléments visibles sont collectés.

Une requête d'extraction se résume donc à la traversée d'un arbre VBSP associé à un calcul 2D d'intersection demi-plan/polygone. Son implantation est immédiate, les calculs minimes et le résultat directement manipulable dans \mathbb{R}^3 . Les parties visibles d'un polygone sont obtenues sous la forme d'une subdivision convexe. Les arêtes de cette subdivision ne délimitent pas nécessairement une partie visible d'une partie masquée. En effet, et comme expliqué dans la section 1.3.3, le calcul de la visibilité dans l'espace de Plücker prolonge les événements visuels le long des droites supportant les arêtes des bloqueurs. De même, les plans calculés par l'algorithme d'extraction prolongent la surface de l'événement le long de la droite support à son arête génératrice. Ce comportement simplifie l'extraction des parties visibles en se ramenant à des opérations élémentaires.

3.2.3 Résultats

L'algorithme que nous venons de présenter a été couplé à un lancer de rayons développé au sein du laboratoire SIC de l'université de Poitiers. En chaque point d'intersection d'un rayon, la requête d'extraction est utilisée pour obtenir la visibilité des sources au point d'impact. Les fragments de sources visibles sont collectés afin de calculer analytiquement l'illumination directe en ce point [NN83, NN85]. Trois scènes de tests sont utilisées. "Grille" (122 polygones, figure 3.10) qui ne comporte qu'une seule source. "Salon" (2206 polygones, figure 3.11) qui comporte deux sources (l'une placée derrière la position de la caméra). Enfin "Classe" (5143 polygones, figure 3.12) qui possède 4 sources. Ces scènes ne sont pas triangulées afin de ne pas démultiplier inutilement le nombre de bloqueurs à traiter pendant le précalcul de la visibilité avec la ou les sources lumineuses. Toutes les images ont pour résolution 1280x960 et sont générées avec un rayon primaire par pixel (ceci pour être cohérent avec les temps de référence calculés). La machine utilisée est un AMD64 3200+ (2.2 GHz) avec 3Go de mémoire.

| Scène | Source(s) | Pc [s] | Nb noeuds | Ombre [s] | Ref [s] | Facteur |
|--------|-----------|-------------|-----------|------------|------------|---------|
| Grille | 1 | 2.0 (2.0) | 1086 | 2.1 (2.1) | 127 (127) | x60 |
| Salon | 2 | 74 (148) | 20981 | 2.5 (5.0) | 444 (888) | x177 |
| Classe | 4 | 135.5 (542) | 93436 | 3.7 (14.8) | 293 (1172) | x80 |

TAB. 3.1 – Comparaison avec un *raytracer* commercial des temps consacrés au calcul des ombres. "Source(s)" indique le nombre de sources lumineuses. "Pc" montre en secondes le temps moyen de précalcul de la visibilité d'une source pour tous les polygones de la scène. "Nb noeuds" est la somme des noeuds des arbres VBSP. "Ombre" présente en secondes le temps total passé par notre approche dans le calcul des ombres. "Ref" correspond aux temps obtenus avec Mental Ray. "Facteur" donne le facteur d'accélération de notre méthode (hors temps de précalcul !). Tous les temps sont donnés en moyenne par source lumineuse. Le temps cumulé pour toutes les sources est indiqué entre parenthèses.

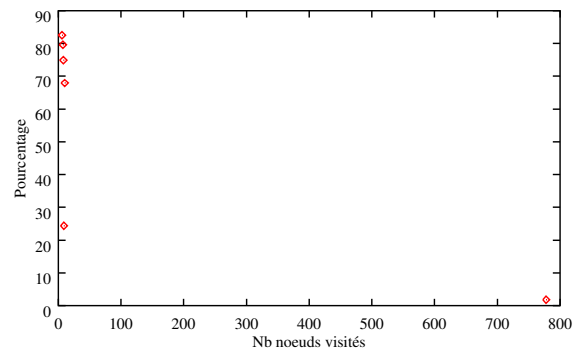
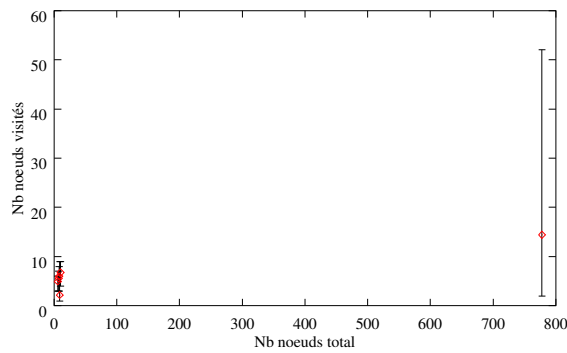
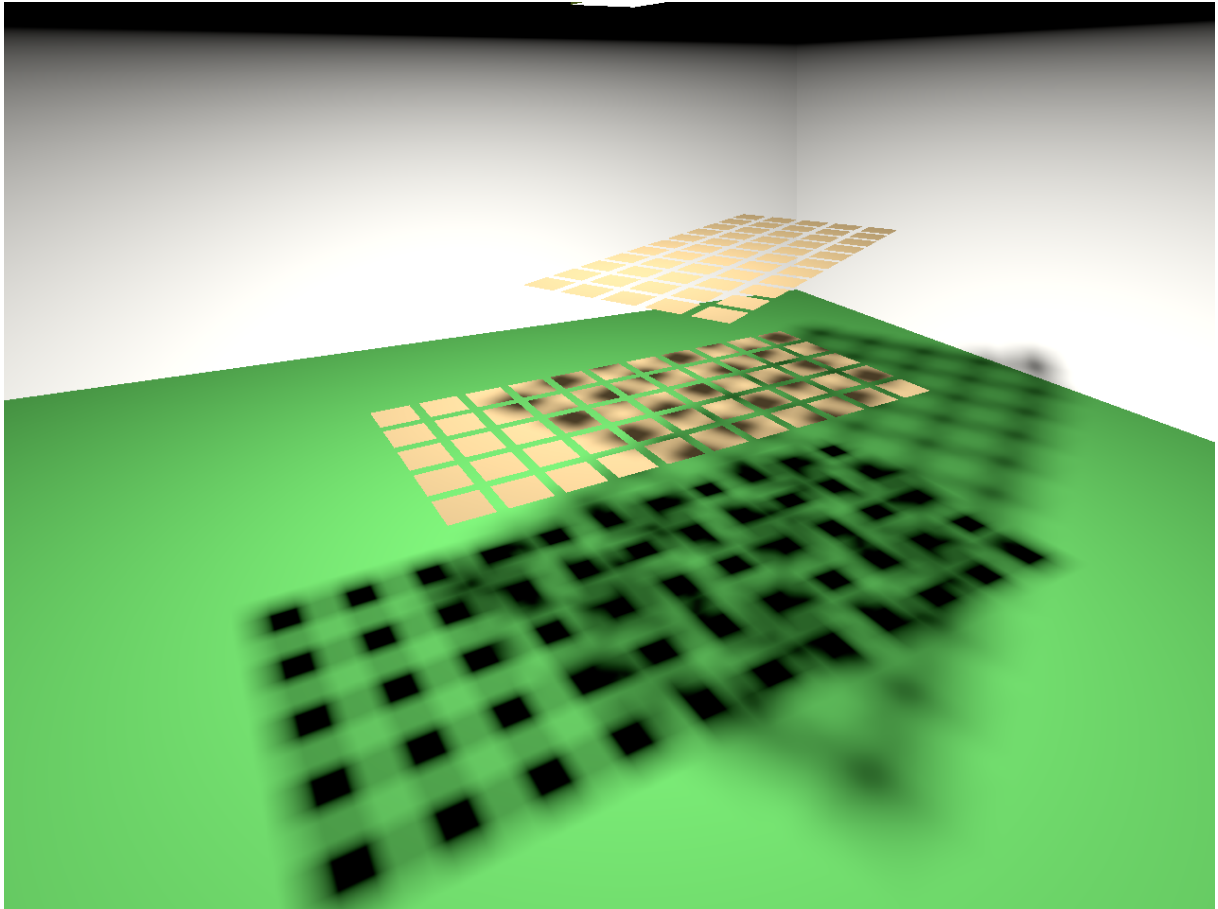


FIG. 3.10 – En haut : la scène “Grille”. En bas à gauche : le nombre moyen de noeuds visités par requête en fonction de la taille des arbres VBSP précalculés. Les barres verticales associées à chaque valeur correspondent à l’intervalle $[min, max]$ des valeurs relevées. En bas à droite : le pourcentage de noeuds visités par requête en fonction de la taille des arbres VBSP.

Les mesures sur l’intervalle $[0, 20]$ correspondent aux polygones de la grille la plus basse ; celle située tout à droite ($n = 778$) correspond aux requêtes effectuées depuis le sol. Dans ce dernier cas, 2% des nœuds suffisent à déterminer la visibilité de la source.

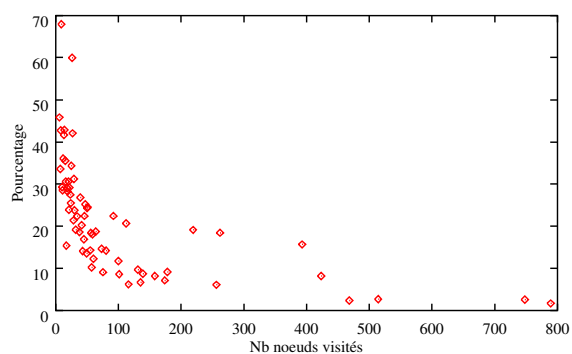
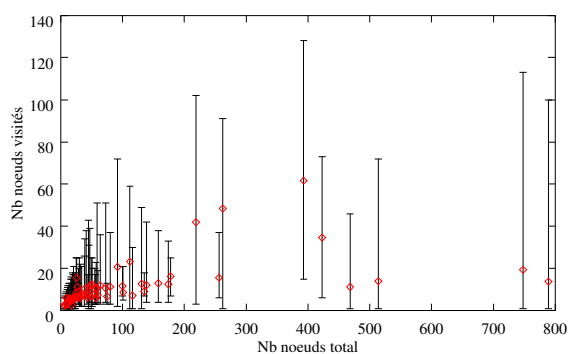


FIG. 3.11 – En haut : la scène “Salon”. En bas à gauche : le nombre moyen de noeuds visités par requête en fonction de la taille des arbres VBSP précalculés. Les barres verticales associées à chaque valeur correspondent à l'intervalle $[min, max]$ des valeurs relevées. En bas à droite : le pourcentage de noeuds visités par requête en fonction de la taille des arbres VBSP.

Les arbres de tailles les plus grandes sont à associer à l'ombre de la plante sur la table ou sur le mur. Les relevés indiquent une réduction du pourcentage de noeuds visités inversement proportionnelle à la taille des arbres.

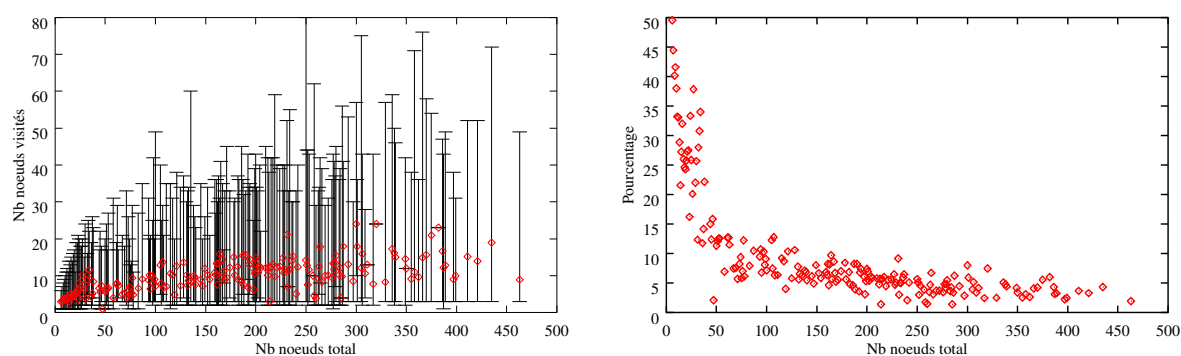


FIG. 3.12 – En haut : la scène “Classe”. En bas à gauche : le nombre moyen de noeuds visités par requête en fonction de la taille des arbres VBSP précalculés. Les barres verticales associées à chaque valeur correspondent à l’intervalle $[min, max]$ des valeurs relevées. En bas à droite : le pourcentage de noeuds visités par requête en fonction de la taille des arbres VBSP.

Le graphe de gauche indique que la complexité pratique de la requête d’extraction tend vers une complexité logarithmique, divergeant clairement de la complexité linéaire théorique.

Temps de rendu

Une première série de tests compare le temps passé au calcul des ombres dans notre logiciel à celui obtenu avec Mental Ray 3.4⁴, un *raytracer* stochastique commercial. Ces temps ont été obtenus en calculant deux fois chaque image, avec et sans ombre, puis en prenant la différence. Le nombre d'échantillons utilisés pour chaque source est de 256. Le tableau 3.1 présente les résultats obtenus. Les trois images issues des rendus présentés dans ce tableau sont celles des figures 3.10, 3.11 et 3.12.

Le gain le plus important est observable pour la scène "Salon" où le facteur d'accélération est de 177. Ce facteur est de 80 pour "Classe" et 60 pour "Grille". Rappelons qu'avant même la réduction des temps de rendu, nos travaux cherchent d'abord à garantir une qualité optimale d'image par rapport au modèle d'illumination fixé. Ces résultats montrent qu'il s'agit d'une exigence conciliable avec un rendu rapide : le précalcul de la visibilité des sources accélère significativement la phase de rendu.

Complexité pratique d'une requête d'extraction

Sous hypothèse qu'un arbre BSP soit équilibré, la complexité de la localisation d'un point est $O(\log n)$ où n est la taille de l'arbre. Si l'objet à localiser n'est pas ponctuel, alors la complexité devient au pire $O(n)$. En pratique, plus l'objet sera petit et plus sa localisation convergera vers une complexité logarithmique. Cette analyse vaut pour une requête d'extraction. Bien que celle-ci opère en 3D, sa complexité est équivalente à localiser dans l'espace de Plücker les régions auxquelles appartient le 3-polytope représentatif des droites passant par un point et une source.

Les figures 3.10, 3.11 et 3.12 présentent le nombre moyen de noeuds visités par requête en fonction de la taille des arbres VBSP. Pour la scène "Grille", les arbres de tailles $n < 20$ correspondent aux polygones de la grille la plus basse. 25 à 85% de leur noeuds sont visités lors d'une requête d'extraction. Ce sont des proportions importantes mais liées à la petite taille des arbres. Les ombres plus complexes sont portées sur le sol dont la visibilité avec la source est représentée par un arbre de taille $n = 778$. Dans ce cas, les requêtes effectuées depuis le sol examinent seulement 2% des noeuds.

Ce comportement se confirme sur la scène "Salon". Les mesures indiquent que le pourcentage de noeuds visités est inversement proportionnel à la taille d'un arbre. Il ne dépasse plus 22% pour $n > 100$.

La scène "Classe" possède le plus grand nombre de mesures. Celles du premier graphe confirment le comportement logarithmique de la requête d'extraction en fonction de la taille des arbres VBSP. De manière générale, l'ensemble des droites passant par un point et une source est petit comparé à l'ensemble total représenté par un arbre VBSP. Ce résultat est conforme à l'analyse faite précédemment, à savoir que la localisation d'un ensemble dans un arbre BSP tend vers une complexité logarithmique lorsque cet ensemble devient petit. Ce comportement de la requête d'extraction explique la rapidité des calculs d'ombres obtenus avec notre méthode.

Augmentation de la taille des sources

La dernière série de tests évalue le comportement de notre algorithme lorsque la taille des sources lumineuses augmente. Pour ce faire, chacune des trois scènes a été calculée 4 fois en multipliant par 2, 4, 6 et 8 la taille des sources (sachant que l'aire initiale des sources est de 400cm^2). Les résultats sont récapitulés dans le tableau 3.2.3 et les images correspondantes listées dans la figure 3.2.3. Quelle que soit la scène, les mesures indiquent un accroissement moyen linéaire du temps de précalcul, de la taille des arbres VBSP et du temps passé à calculer les ombres.

Sur la scène "Grille", une régression linéaire donne des pentes de 0.35 pour ce qui est du temps de précalcul par source, 176.7 pour le nombre total de noeuds, 0.49 pour le temps de calcul des ombres

⁴<http://www.mentalimages.com/>

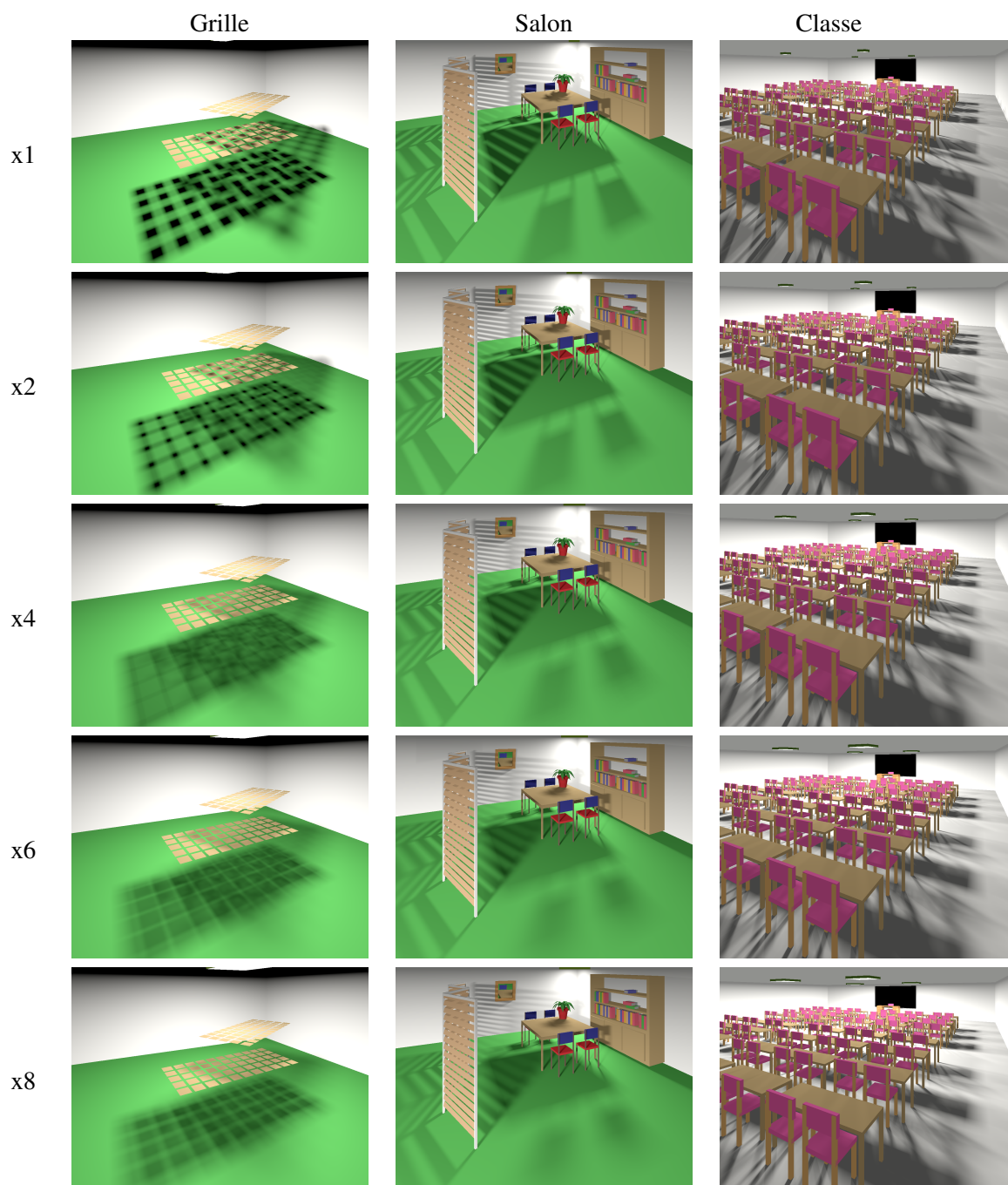


FIG. 3.13 – Les trois scènes de tests calculées avec des sources de superficies croissantes.

par source. Pour la scène “Salon”, ces valeurs sont respectivement de 29, 4433.3, 0.21. Pour la scène “Classe”, elles sont de 31.4, 12731, 0.16. On peut remarquer que l’augmentation du temps de calcul des ombres est la plus forte sur “Grille”, ensuite sur “Salon”, et la plus faible sur “Classe”. Les requêtes les plus coûteuses sont celles effectuées sur les arbres VBSP les plus grands. Or, sur “Classe” la taille maximale des arbres est inférieure à celles de “Salon” ou “Grille”. Sur cette dernière, l’arbre le plus grand correspond au sol qui occupe une part importante de l’image. De ce fait, le nombre de requêtes faites à

| Scène | Source(s) | Taille | Pc [s] | nb noeuds | Ombre [s] |
|--------|-----------|--------|--------------|-----------|------------|
| Grille | 1 | x1 | 2.0 | 1086 | 2.1 |
| | | x2 | 2.4 | 1380 | 2.5 |
| | | x4 | 3.2 | 1731 | 3.7 |
| | | x6 | 3.9 | 2073 | 4.5 |
| | | x8 | 4.4 | 2348 | 5.5 |
| Salon | 2 | x1 | 74 (148) | 20981 | 2.5 (5.0) |
| | | x2 | 113 (226) | 26588 | 3.1 (6.3) |
| | | x4 | 177 (354) | 36270 | 3.6 (7.2) |
| | | x6 | 230 (460) | 45026 | 3.8 (7.6) |
| | | x8 | 278 (556) | 51908 | 4.1 (8.2) |
| Classe | 4 | x1 | 135.5 (542) | 93436 | 3.7 (14.8) |
| | | x2 | 170 (680) | 110666 | 3.9 (15.6) |
| | | x4 | 234 (936) | 138049 | 4.2 (16.8) |
| | | x6 | 296.5 (1186) | 162220 | 4.6 (18.4) |
| | | x8 | 355 (1420) | 183066 | 4.8 (19.2) |

TAB. 3.2 – Les trois scènes de tests calculées avec la superficie des sources multipliée par 2, 4, 6 et 8. (x1 = 400cm²). Comme pour les précédentes mesures, tous les temps en secondes sont des valeurs moyennes par source, le temps cumulé pour toute les sources étant indiqué entre parenthèses. La colonne “nb noeuds” est le nombre total de noeuds de tous les arbres VBSP calculés.

partir du sol est élevé. Ceci explique que l'augmentation des temps de calculs soit la plus importante sur cette scène, bien qu'elle soit la plus petite des trois.

3.3 Discussion et perspectives

Les travaux présentés dans ce chapitre montrent que le calcul analytique de l'illumination directe peut être effectué en tout point d'une image sans sacrifier la vitesse du rendu. La qualité des ombres calculées est optimale vis-à-vis du modèle d'illumination choisi. L'efficacité de notre approche repose d'une part sur le précalcul de la visibilité des sources, et d'autre part sur la vitesse d'exécution d'une requête d'extraction. Contrairement aux autres méthodes, le précalcul de visibilité des sources prend en compte l'occultation, y compris celle liée à la conjonction de multiples bloqueurs. En outre, la représentation hiérarchique de la visibilité par arbre BSP induit un comportement logarithmique de la requête d'extraction.

3.3.1 Extensions et optimisations

Dans ce chapitre, nous avons fait l'hypothèse que toutes les sources étaient d'émission uniforme, permettant le calcul analytique de leurs contributions en un point de la scène. L'extension à des modèles de sources plus complexes est cependant possible : après avoir obtenu une subdivision convexe d'une source, chaque partie visible peut être échantillonnée afin d'établir sa contribution en un point selon une approche type Monte-Carlo.

La transparence est une propriété qui n'est pas actuellement supportée par notre implantation. Cependant, il s'agit là d'une extension réalisable. Une solution est d'effectuer le calcul des arbres VBSP en ignorant dans un premier temps les bloqueurs transparents. Ensuite, chaque bloqueur transparent est inséré dans l'arbre. Au lieu d'éliminer les parties communes à l'arbre et au bloqueur, celles-ci demeurent en étant marquées d'une étiquette indiquant qu'une source est visible à travers un objet. Lors d'une requête d'extraction, cette modification permettrait d'obtenir la liste des bloqueurs transparents à travers lesquels une source ou une partie de source est visible.

Par ailleurs, les temps de rendu et de précalcul présentés peuvent encore être améliorés. Pour notre approche, diminuer le temps de calculs des ombres passe par une réduction de la taille des arbres VBSP. Autrement dit, l'étape de précalcul doit être réalisée en utilisant moins d'hyperplans, *i.e.* moins d'arêtes de bloqueurs. De fait, la restriction du précalcul de la visibilité aux seules arêtes silhouettes des objets est une optimisation possible. Rappelons que la silhouette d'un objet est moins complexe que le maillage de polygones qui le compose. L'étude réalisée par McGuire [McG04] illustre cette propriété⁵. De plus, l'intérêt du procédé a déjà été mis en avant par Haumont *et al.* [HMN05]. Bien qu'il s'agisse uniquement de prouver la visibilité, les résultats obtenus démontrent un gain significatif en temps lorsque les silhouettes sont utilisés.

3.3.2 Limitations actuelles

Temps de précalcul

Notre approche étant indépendante de la position de la caméra, les temps de précalcul ne sont pas pris en compte pour établir les facteurs d'accélération du tableau 3.1. Clairement, les travaux proposés dans ce chapitre s'inscrivent parmi les méthodes dites *out-of-core*, tirant partie d'une information précalculée. En ce sens, le temps consacré au précalcul de cette information est d'une importance secondaire. Cependant, il ne doit pas devenir prohibitif pour autant. Le temps de précalcul est linéaire par rapport au nombre de polygones et de sources dans la scène. Ce comportement est déjà très net sur les 3 scènes utilisées, qui sont pourtant de taille modeste. Cette complexité du précalcul peut devenir un facteur limitant

⁵ Etude réalisée sur la silhouette d'objets depuis des points de vue. Les résultats sont donc à remettre en perspective vis-à-vis de l'extension de la notion de silhouette à une source surfacique

sur des scènes plus importantes.

Pour la casser, une solution classique est de hiérarchiser le précalcul. Le procédé est appliqué par Bittner [Bit02], en particulier pour les arbres d'occultation. Dans notre cas, il s'agit de calculer hiérarchiquement la visibilité d'une source avec les cellules d'une subdivision spatiale, afin de trouver les régions qui lui sont totalement masquées. On évite ainsi d'effectuer le calcul pour chacun des polygones contenus. Une approche similaire est d'ailleurs présentée dans le chapitre 4. Nous ne l'avons pas utilisée ici car la taille des scènes ne l'imposait pas. De plus, cette solution est principalement efficace pour des environnements où l'occultation est forte, dans le cas de scènes architecturales comportant plusieurs pièces par exemple.

Une solution alternative est de calculer la visibilité d'une source pour un ensemble de polygones. Comme indiqué dans la section 3.2.1, figure 3.8, la visibilité d'une source depuis un polygone est aussi la visibilité de cette source *à travers* ce polygone. En calculant la visibilité depuis les faces d'une région convexe, la requête d'extraction est applicable en tout point de son intérieur. On peut donc utiliser la visibilité de la région, pour extraire la visibilité des polygones qu'elle contient. De plus, si les régions considérées sont celles d'une subdivision spatiale, cette méthode se combine parfaitement avec une hiérarchisation du précalcul évoquée précédemment. A ceci, on objectera qu'il faut pouvoir prendre en compte l'occultation interne à la région. Cela fait partie des développements que nous souhaitons donner aux travaux présentés dans ce chapitre.

Robustesse

Une seconde limitation est le problème de la robustesse. Basée sur des calculs d'intersection 2D droite/polygone, la requête d'extraction est très robuste. Mais la validité de son résultat dépend de la validité du précalcul. Dans le précédent chapitre, nous avons déjà souligné que l'algorithme de Bajaj *et al.* [BP96] pouvait être pris en défaut si des erreurs étaient commises dans la classification des sommets d'un polytope par rapport à un hyperplan (voir la section 2.3.3). Cette configuration peut survenir lorsque les objets occultants sont formés d'un maillage fin de polygones. De fait, ce n'est pas tant le nombre d'objets à traiter qui pose problème que le niveau de détail avec lequel ils sont modélisés. L'échec d'un précalcul de visibilité entre un polygone et une source obligerait à reléguer à un autre algorithme (tel que [TT97] par exemple) l'extraction des parties visibles de la source depuis le polygone incriminé.

En ce sens, baser le précalcul des visibilitées sur la silhouette des objets est également une réponse au problème de robustesse. Cela permet de ne pas devoir traiter l'intégralité d'un maillage et de réduire considérablement le nombre d'opérations CSG à effectuer dans l'espace de Plücker. En calculant moins d'intersections, le risque d'être confronté à des problèmes d'instabilité numérique est réduit. Rappelons que l'algorithme de Bajaj et Pascucci [BP96] minimise déjà les calculs numériques. Ce n'est donc pas sur l'algorithme d'intersection que l'on gagnera significativement en robustesse, mais bien sur le nombre d'appels à cet algorithme.

3.3.3 Perspectives

Visibilité absolue

On montre facilement qu'un arbre VBSP associé à deux polygones A et B supporte des transformations géométriques telles que les rotations, les translations (toute isométrie affine en général) et les changements d'échelles conservant les proportions. Si ces transformations sont appliquées à la fois à A et B et à leurs bloqueurs, il suffit de les répercuter sur les noeuds de l'arbre pour que celui-ci demeure valide.

Par visibilité absolue d'un objet, nous entendons l'intersection des droites de \mathbb{R}^3 avec l'objet en question.

En ce sens, on peut le voir comme une généralisation de l'asp [Pla88] d'un polygone. Cette visibilité est "absolue" puisqu'elle ne dépend d'aucun point de vue (qu'il s'agisse d'un point, d'une face, ou d'une région d'observation). Considérons la boîte englobante d'un objet. Si, pour chaque couple distinct de faces de la boîte, un arbre VBSP est calculé, l'ensemble des arbres VBSP est une représentation des droites qui intersectent l'objet contenu.

L'intérêt d'une telle représentation est que si l'objet est déplacé, tourné, redimensionné, alors sa visibilité absolue n'a pas à être recalculée. De plus, on retrouve logiquement des propriétés similaires à celles de l'asp [PD90] (voir aussi 1.4.3) : l'occultation engendrée par plusieurs objets est l'union de leur visibilité absolue, et l'occultation qu'engendre un objet sur un autre est la différence de leurs visibilités absolues. L'algorithme d'extraction présenté dans ce chapitre demeure alors valide. L'ensemble des droites intersectant un objet est l'occultation qu'il engendre, quel que soit le point d'observation. Dès lors, depuis un point w , déterminer les parties visibles d'un polygone partiellement masqué par un objet consiste à "filtrer" ce polygone à travers les arbres VBSP associés à cet l'objet. Ceci se généralise à plusieurs objets occultants. Dans ce cas, les fragments visibles de ce polygone doivent être successivement filtrés à travers les arbres VBSP de chaque objet.

Une telle approche, basée objet, ouvre plusieurs perspectives. Actuellement, tout changement dans la géométrie de la scène nécessite d'effectuer à nouveau le calcul des arbres VBSP, au moins pour les couples (source, polygone) dont la visibilité est affectée par la modification. En se basant sur la notion de visibilité absolue, seule celle de l'objet modifié, autrement que par une isométrie, devra être recalculée. Cette approche permettra également de proposer une architecture différente, adaptée des travaux de Lext *et al.* [LAM01] ou de Wald *et al.* [WBS03] pour le rendu de scènes incluant des objets en mouvement. Leurs solutions consistent à tenir à jour un kd-tree global construit uniquement en utilisant les boîtes englobantes des objets contenus. Chaque objet est associé à un kd-tree local, *i.e.* un kd-tree restreint au volume de sa boîte englobante. De la sorte, la méthode supporte le déplacement d'objets : seul le kd-tree global nécessite d'être recalculé, chaque kd-tree local est inchangé. À terme, nous envisageons une démarche similaire en associant à chaque objet sa visibilité absolue. En rendu interactif, *a fortiori* pour des scènes dynamiques, la prise en compte des ombres douces fait chuter considérablement les performances, et demeure problématique. La méthode que nous avons présentée pourrait constituer une solution à cet écueil.

Illumination globale

La généralisation de notre algorithme à des calculs d'illumination globale fait également partie des perspectives à venir. Des tests préliminaires encourageants ont été effectués sur une approche Monte-Carlo utilisant notre algorithme uniquement pour le calcul de l'illumination directe. Les développements futurs reposent aussi sur la notion de visibilité absolue. Si le procédé décrit précédemment permet l'extraction de la visibilité des sources, il peut être facilement étendu pour extraire la visibilité de n'importe quel polygone de la scène depuis n'importe quel point donné.

L'opportunité est donnée de généraliser l'approche suivie par Durand *et al.* [DDP98, DDP99] afin de guider par la visibilité la propagation de l'énergie dans la scène. De plus, tout facteur de forme pourra être obtenu avec précision en tout point de la scène, et non pas uniquement aux sommets des polygones.

Prédiction sans perte de la propagation des ondes électromagnétiques

Sommaire

| | | |
|------------|--|------------|
| 4.1 | Préambule | 87 |
| 4.1.1 | Les enjeux | 87 |
| 4.1.2 | Caractéristiques du canal de propagation | 87 |
| 4.1.3 | Les modèles basés rayons | 88 |
| 4.2 | Approche proposée | 92 |
| 4.2.1 | Requête de visibilité unifiée | 92 |
| 4.2.2 | Graphe de visibilité : calcul hiérarchique mixte | 99 |
| 4.2.3 | Résultats | 104 |
| 4.3 | Conclusions | 108 |
| 4.3.1 | Discussion | 108 |
| 4.3.2 | Perspectives | 109 |

Ce chapitre présente une optimisation sans perte pour la simulation de la propagation des ondes électromagnétiques, utilisant un modèle basé rayons.

Après avoir présenté les enjeux inhérents à ce type d'application, nous expliquons en quoi les modèles de propagation basés rayons se démarquent avantageusement des autres modèles. Cependant, ces modèles sont énormément consommateurs de temps CPU et ne sont pas utilisables sans optimisation. A notre connaissance, toutes les optimisations existantes altèrent la fiabilité des résultats.

De cette analyse, nous proposons une optimisation garantissant une fiabilité de résultat optimale pour le modèle choisi. Notre approche utilise un graphe encodant les relations de visibilité entre les polygones et les arêtes d'un environnement donné. Son calcul s'appuie sur un algorithme hiérarchique, combinant différentes requêtes de visibilité adaptées à la précision requise. En particulier, nous verrons comment utiliser le complexe de polytopes associé à deux polygones pour extraire la visibilité de leurs arêtes.

L'évaluation de notre approche indique qu'elle permet l'utilisation sans perte de modèles basés rayons sur des scènes significatives tout en conservant des temps praticables. A notre connaissance, il n'existe

pas d'autre méthode satisfaisant les mêmes propriétés.

Nous discutons en fin de chapitre des limitations actuelles de notre approche et des développements futurs que nous souhaitons leur donner.

Les travaux présentés dans ce chapitre ont été développés dans le cadre du contrat n° 42 538 233 entre le laboratoire SIC et France Telecom R&D Belfort.

4.1 Préambule

4.1.1 Les enjeux

Ces dernières années, les télécommunications sans fil ont connu un essor sans précédent. Les progrès techniques les ont rapidement rendues accessibles au grand public. Il s'est créé un marché où la concurrence est à la hauteur de la formidable manne économique qu'il représente. Au delà de toute considération commerciale, c'est avant tout une guerre technologique qui se joue, motivée par l'augmentation des volumes de données numériques à transmettre. Car il ne s'agit plus seulement de voix aujourd'hui, mais aussi de courrier électronique et de services en ligne consommateurs de bande passante tels que la musique, la photo ou la vidéo. La transmission haute qualité de ces données nécessite la mise en oeuvre de systèmes sans fil toujours plus performants et disposant d'une bande passante toujours accrue. A ceci s'ajoute aujourd'hui le développement des réseaux informatiques sans fil, type WIFI et WLAN. Leur souplesse d'utilisation et leur infrastructure simplifiée viennent compenser des performances pour le moment encore inférieures aux réseaux classiques.

Pour concevoir les prochaines générations de systèmes de télécommunications, un enjeu majeur est la maîtrise du canal de propagation. Il s'agit de l'ensemble des éléments qui affectent une transmission entre un émetteur et un récepteur. Cela inclut principalement la nature de la densité de l'urbanisme. Puis, la spécificité des applications (réseaux ad-hoc, WLAN) peut nécessiter de prendre en compte des éléments supplémentaires, tels que la mobilité du récepteur, ou bien le mobilier présent dans l'environnement de propagation. L'amélioration des systèmes de télécommunications sans fil passe ainsi par une connaissance toujours plus précise du canal de propagation. Différents paramètres ont été établis pour le caractériser. Ce sont des indicateurs essentiels qui permettent, par exemple, d'améliorer le déploiement des antennes ou de prédire la qualité de la réception.

4.1.2 Caractéristiques du canal de propagation

Toute l'information permettant de décrire un canal de propagation est contenue dans la *réponse impulsionnelle*. Considérons l'image d'une personne éternuant dans une église. L'onde sonore provoquée, par réverbération sur les murs ou les colonnes, engendre des ondes sonores secondaires. Selon la nature et l'emplacement des interactions subies, le cheminement de chaque onde est différent. De fait, elles reviennent aux oreilles de la personne espacées dans le temps, avec plus ou moins d'intensité. Autrement dit, c'est le phénomène d'écho. Ici, la réponse impulsionnelle correspond au relevé de la puissance de chaque onde sonore à l'instant où elle est perçue.

De manière simpliste, transmettre une information, c'est l'encoder en un ensemble de paquets élémentaires, ou *symboles*, émis successivement. Par interaction avec l'environnement, comme pour l'écho, un symbole émis une fois peut parvenir à un récepteur plusieurs fois avec une puissance variable et à des instants différents. En radiocommunication, la réponse impulsionnelle est le relevé en puissance en fonction du temps de propagation des ondes reçues en un point de réception donné, pour une position d'émetteur fixée. A partir de la réponse impulsionnelle, toutes les caractéristiques d'un canal de propagation peuvent être obtenues. On citera principalement :

L'atténuation en puissance, qui définit la déperdition moyenne d'énergie électromagnétique entre l'émetteur et une position de récepteur donnée. Cette caractéristique est d'importance pour le déploiement d'une antenne, où l'on cherche à maximiser sa zone de couverture, *i.e.* la zone pour laquelle la puissance moyenne reçue est suffisante pour établir une communication. Plus cette puissance est faible, plus la communication risque d'être bruitée, ce qui implique une baisse du débit en raison d'erreurs de transmission.

La dispersion des retards donne une mesure de la durée durant laquelle un même symbole peut être reçu. C'est un indicateur de la qualité de la réception. En particulier, il permet de prédire l'apparition d'interférences entre symboles. Car si la dispersion des retards est supérieure à la durée séparant l'émission de deux symboles, le second peut parvenir au point de réception alors que le premier continue d'arriver. Pour conserver l'image de l'écho, si une personne parle, les mots qu'elles prononcent peuvent lui revenir plusieurs fois, éventuellement en se superposant, engendrant ainsi un effet de bruit. C'est exactement le même problème en radiocommunication et on peut noter que le même phénomène est à l'origine des interférences lumineuses.

Ce n'est là qu'un bref aperçu des caractéristiques du canal de propagation et des problématiques qui lui sont liées. Cependant, il montre déjà qu'une connaissance du canal de propagation est précieuse pour la mise en œuvre d'un système de radiocommunication. Encore faut-il pouvoir en prédire les caractéristiques. Pour cela, il n'existe que deux possibilités : effectuer des campagnes de mesures, ou simuler la propagation des ondes.

La mesure donne accès à la réponse impulsionnelle, mais possède des inconvénients majeurs. Tout d'abord, elle est lente et très coûteuse. Lente, car une équipe technique doit quadriller toute une zone (qui peut être une ville) en effectuant des relevés à intervalles réguliers, ce qui peut prendre des mois. Coûteuse, car il faut financer cette équipe et la doter d'un matériel de mesure très onéreux. Ensuite, la mesure a ses limites. Par exemple, le déploiement d'une antenne doit optimiser la zone qu'elle peut couvrir, mais la mesure implique que l'antenne soit déjà en place... De plus, une campagne de mesures possède une longévité limitée en raison de l'évolution de l'urbanisme (destructions ou constructions de bâtiments). Enfin, un relevé de mesures n'est jamais parfait. C'est une image de la réalité sensiblement altérée par les conditions expérimentales (densité du trafic routier par exemple) et par les imprécisions dues au matériel de mesure utilisé.

Afin de pallier ces problèmes, différents modèles de propagation des ondes électromagnétiques [IY02] ont été développés. Sur le plan théorique, tout phénomène électromagnétique est régi par les équations de *Maxwell*. Tout modèle de propagation peut être vu comme une résolution plus ou moins approchée de ces équations. La capacité d'un modèle à converger vers leur solution détermine sa précision, et conditionne son champ d'application. Dans les deux cas, les modèles basés rayons s'imposent.

4.1.3 Les modèles basés rayons

Fondements et validité

La propagation d'une onde électromagnétique est régie par les équations de *Maxwell* [Cha97, BW97]. En milieu homogène et isotrope (l'air est considéré comme tel), l'équation de *Helmoltz* donne une solution générale des équations de *Maxwell*, et décrit la propagation d'une onde sinusoïdale. Les modèles basés rayons dérivent de la résolution par développement limité de l'équation de *Helmholtz*. La série de *Luneberg-Kline* [BM94] est une solution de cette équation. En particulier, son premier terme donne une preuve de l'Optique Géométrique, et valide l'approche rayon dans un milieu homogène et isotrope. Il justifie les notions de réflexions spéculaires et de réfraction (ou transmission) au niveau d'une interface séparant deux milieux. Le phénomène de diffraction, pris en compte par la Théorie Géométrique de la Diffraction (ou TGD), se retrouve dans les termes suivants de la solution de *Luneberg-Kline*. La TGD est une extension de l'Optique Géométrique proposée par Keller [Kel62] afin de prendre en compte la diffraction. Cette théorie, et celles qui suivirent (Théorie Uniforme de la Diffraction [KP74]) sont très importantes en radiocommunication où la diffraction joue un rôle essentiel. En effet, un rayon incident à une arête est démultiplié en un ensemble de rayons diffractés. Le principe de Fermat généralisé, énoncé par Keller, précise que ces rayons sont portés par un cône (le cône de Keller) dont les caractéristiques

dépendent de l'angle d'incidence du rayon.

Sur ces fondements théoriques, les modèles basés rayons supportent tous les mécanismes inhérents à la propagation d'une onde. Cependant, ils ne sont pas pour autant exacts, et sont confrontés à deux types de problèmes, l'un théorique, l'autre pratique.

Théoriquement, l'interaction d'une onde avec un élément du milieu de propagation n'est valide que si cet élément a des dimensions supérieures à la longueur d'onde du signal. Par exemple, il n'est pas possible de considérer des objets de taille inférieure ou égale à 30 cm pour un signal dont la fréquence est 1GHz. Aussi, les objets de dimensions inférieures à la longueur d'onde utilisée sont traditionnellement ignorés. Enfin, en pratique, l'utilisation d'un modèle basé rayons implique une connaissance précise de l'environnement de propagation. Cela concerne sa géométrie, mais aussi les propriétés électromagnétiques des matériaux en présence. Dans les deux cas, ces informations sont difficiles à obtenir avec précision. Par exemple, la géométrie d'une ville est issue des données de l'Institut Géographique National dont les mesures sont faites au mètre près. Quant à la nature des matériaux, c'est une information encore plus dure à obtenir, *a fortiori* à grande échelle.

Quoi qu'il en soit, les modèles basés rayons sont très utiles en radiocommunication. Ils sont réputés fiables et, parmi les modèles utilisables sur des scènes de tailles significatives, ils demeurent les plus performants en terme de précision. C'est aussi ce qui les rend intéressants sur un plan industriel.

Approches existantes

La majorité des techniques utilisées en radiocommunication sont issues de la synthèse d'images. Ce ne sont pas les plus récentes, compte tenu des années nécessaires au transfert d'un savoir d'un domaine de recherche à un autre. De manière générale, on trouve trois types d'approches : les modèles basés lancer de rayons, les modèles basés source-images, et les modèles hybrides.

Les modèles basés lancer de rayons sont les plus répandus [LCL89]. Depuis la position de l'émetteur, les rayons sont lancés dans toutes les directions selon un pas angulaire fixe. Chaque polygone atteint génère un rayon réfléchi et/ou transmis, calculé en fonction des propriétés électromagnétiques de la surface. Chaque arête atteinte diffracte le rayon, *i.e.* le point d'incidence est alors considéré comme un nouvel émetteur secondaire. Chaque récepteur atteint ajoute à sa réponse impulsionnelle le trajet qui lui est parvenu. En raison du pas angulaire, l'approche est sujette aux problèmes de sous échantillonnage : toute la scène n'est pas couverte et des trajets sont perdus.

Pour pallier ce problème, le lancer de cônes [SR94] ou de tubes [SM97] ont été introduits. Ces méthodes s'apparentent aux travaux de Amanatides *et al.* [Ama84] ou de Ghazanfarpour *et al.* [GH98] en synthèse d'images (voir la section 3.1.4). L'objectif ici est de garantir la couverture complète de la scène. Répandu, le lancer de cônes introduit des problèmes de recouvrement du champ électromagnétique, lorsqu'un récepteur est à l'intersection de deux cônes. Un même trajet peut être comptabilisé deux fois et fausser le calcul de la réponse impulsionnelle. Des corrections doivent être apportées [YIZ01].

La complexité de cette approche est linéaire en fonction du nombre total de rayons lancés dans la scène, ce qui dépend du nombre maximum d'interactions autorisées pour un trajet. En particulier le nombre de diffractions est dominant puisque cela revient à considérer des émetteurs secondaires quand une réflexion ou réfraction n'engendre qu'un unique rayon. Néanmoins, les modèles basés lancer de rayons sont rapides et permettent de traiter des scènes à l'échelle de grandes agglomérations. En revanche, ils ne permettent pas de déterminer tous les chemins entre un émetteur et un récepteur. Le choix d'un pas angulaire implique une perte de trajets qui se traduit par autant d'erreurs dans le calcul des réponses impulsionnelles.

Les modèles basés source-images exploitent la nature spéculaire des réflexions. La figure 4.1 illustre le

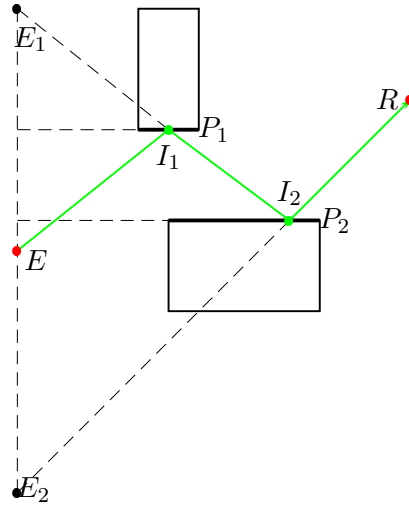


FIG. 4.1 – Principe de la méthode des source-images

principe de la méthode des source-images, une application directe de la loi de Descartes. Le symétrique de l'émetteur E par rapport au plan support du polygone P_1 définit la première source-image E_1 de E . De même, E_2 est une nouvelle source-image par symétrie de E_1 par rapport à P_2 . La droite formée par E_2 et le récepteur R intersecte P_2 et définit ainsi le point de réflexion I_2 . De même, $(E_1 I_2)$ intersecte P_1 au point de réflexion I_1 . Le trajet $\{E, I_1, I_2, R\}$ est ainsi déterminé. Le même raisonnement s'étend à la transmission.

Lorsqu'une diffraction a lieu, la même approche est appliquée au point diffractant considéré comme un émetteur secondaire. De la sorte, pour un nombre donné d'interactions, toutes les contributions entre un émetteur et un récepteur sont trouvées. Cette technique n'est sujette à aucun problème de sous-échantillonnage et peut être ainsi considérée comme la seule exacte pour un modèle basé rayons. Cependant, la recherche exhaustive des trajets est soumise à une complexité combinatoire qui rend son utilisation délicate en pratique. Pour un émetteur et un récepteur dans une scène de n polygones et arêtes, la complexité est en $O(n^{(I_d+I_r+I_t)})$ où I_d , I_r , I_t sont respectivement les nombres de diffractions, de réflexions et de transmissions autorisés. La méthode devient rapidement impraticable lorsque ces valeurs augmentent, même si n est faible. Pour cette raison, les modèles basés source-images sont très rares. Le simulateur *CRT (Communication Ray-Tracer)* [Ave99, APVM00] développé au laboratoire SIC de l'université de Poitiers en fait partie.

Les méthodes hybrides sont des modèles basés lancer de rayons dont la précision est améliorée par combinaison avec le principe des source-images [TT96]. Dans le cas d'un lancer de rayons, la probabilité qu'un rayon atteigne exactement le récepteur est infinitésimale. Une boule centrée sur sa position est utilisée et tous les rayons qui l'intersectent sont comptabilisés. Les trajets considérés ne sont pas exactement ceux qui atteignent le récepteur. Des imprécisions sont donc commises sur la réponse impulsionnelle.

Pour éliminer ce problème, les modèles hybrides mémorisent pour chaque trajet la séquence des objets rencontrés. Lorsqu'un rayon atteint la boule d'un récepteur, la méthode des source-images est utilisée sur cette séquence pour déterminer si un trajet le relie exactement à l'émetteur. Si tel est le cas la contribution électromagnétique est précisément déterminée.

Une approche hybride tire profit de la précision des source-images et bénéficie de la rapidité du lancer de rayons. Si elle ne permet pas pour autant de trouver tous les trajets, elle améliore néanmoins la fiabilité des résultats.

Dans le domaine des radiocommunications, diverses optimisations sont souvent utilisées pour accélérer la vitesse des modèles. Il s'agit principalement de grilles régulières pour optimiser la recherche du premier objet intersecté par un rayon. Les structures hiérarchiques de type *octree* ou *kd-tree* ne sont pas encore très répandues.

Une autre technique issue de la synthèse d'images est le *Z-buffer angulaire* [CJP98, dABD⁺00], qui correspond à un *light buffer* [HG86] (voir la section 3.1.3). Un *Z-buffer angulaire* est calculé pour chaque source (émetteur, source-image ou point de diffraction) afin d'accélérer le lancer des rayons émis. Néanmoins, la démultiplication des sources limite l'application de la méthode.

Enfin, certains travaux observent qu'une condition nécessaire à l'existence d'un trajet est que les objets constituant sa séquence soient visibles deux à deux. Les scènes étant statiques, ces relations de visibilité sont invariantes et peuvent être calculées une fois puis stockées dans un fichier. Pour des modèles basés lancer de rayons, cette information est utilisée pour rechercher ou même prédéfinir les intersections rayons/polygones [AFF97, HWL99, AFR⁺00]. Dans le cas d'un modèle basé source-images, elle permet de restreindre la recherche combinatoire de trajets aux seuls couples de primitives visibles [MA05]. À notre connaissance, toutes ces techniques introduisent actuellement des pertes de trajet en raison de la nature (très) agressive des calculs de visibilité réalisés. Par exemple, Hoppe *et al.* [HWL99] subdivisent la géométrie de la scène en patchs et précalculent la visibilité de deux patchs en testant la visibilité de leur barycentre. Sachant que la superficie d'un patch est de 2 à 5 mètres carrés, le procédé peut être qualifié de hautement agressif.

Problématique

Comme indiqué au début de cette section, les fondements théoriques d'un modèle basé rayons ne permettent pas de prendre en compte les interactions avec les objets de taille inférieure à la longueur d'onde du signal. De plus, la précision du modèle est nécessairement corrélée à la précision de la modélisation de l'environnement de propagation. De fait, comme tout modèle, un modèle basé rayon demeure une approximation de la propagation des ondes électromagnétiques. Les résultats qu'ils permettent d'obtenir sont donc sujets à vérification. Pour ce, les valeurs calculées à l'aide d'une technique donnée sont traditionnellement comparées à des séries de mesures, afin d'établir la validité de la méthode. C'est une étape classique avant son utilisation à des fins industrielles par exemple.

Il faut souligner qu'à ce jour, toutes les méthodes *praticables* accentuent les approximations inhérentes aux modèles basés rayons, puisqu'elles introduisent des pertes de trajets. Par nature, toutes les approches utilisant un lancer de rayons ne peuvent être exactes vis-à-vis du modèle. Si la méthode des source-images est initialement sans perte, elle est inutilisable sur des environnements de taille significative, à moins de recourir à des optimisations qui introduisent toutes des pertes [AC04].

Le potentiel des modèles basés rayons demeure ainsi sous-exploité puisqu'aucune solution praticable ne permet de garantir une précision optimale vis-à-vis du modèle. En outre, sans méthode exacte et praticable, il est impossible d'évaluer véritablement l'erreur commise par une solution approchée.

L'objectif des travaux présentés dans ce chapitre est précisément de combler ce vide. Il s'agit d'optimiser un modèle basé source-images afin de le rendre praticable sur des environnements de taille significative tout en garantissant l'absence de perte de trajet.

4.2 Approche proposée

Cette section présente notre méthode de simulation de la propagation des ondes électromagnétiques. L'objectif est de produire un outil qui trouve exactement tous les trajets entre un émetteur et un récepteur, en conservant des temps de calculs praticables sur des environnements de taille significative. De la sorte, cet outil sera la première véritable alternative à la mesure.

Pour développer ces travaux, nous nous plaçons dans des environnements extérieurs 2.5D, avec comme contrainte de pouvoir facilement évoluer vers des environnements 3D.

Le principe de notre approche est de coupler un modèle basé source-images à un graphe exact encodant toutes les relations de visibilité entre toutes les primitives de la scène. Ces primitives sont soit les polygones sur lesquels des réflexions¹ peuvent avoir lieu, soit leurs arêtes sur lesquelles des diffractions peuvent apparaître. Pour un environnement donné, un graphe est calculé une fois et stocké dans un fichier. Pour calculer tous les trajets entre un émetteur E et un récepteur R , l'algorithme combinatoire de recherche est restreint aux couples de primitives visibles. De par la nature exacte des calculs de visibilité, le procédé garantit de ne perdre aucun trajet.

Dans le pire des cas, *i.e.* lorsque tous les couples de primitives sont visibles, la complexité de notre approche est aussi prohibitive qu'une recherche combinatoire *brute force*. Cependant, une telle configuration relève du cas (très) particulier, et l'on peut s'attendre à une complexité moindre en pratique, comme le montreront les résultats.

Notre méthode nécessite de satisfaire à deux problématiques. La première est qu'il faut prouver la visibilité de deux arêtes, d'une arête et d'un polygone, et de deux polygones. En effet, le phénomène de diffraction implique la génération de rayons diffractés par des arêtes. Leur visibilité est donc à prendre en compte. Cela correspond à trois types de requêtes différents. La seconde problématique est que ces requêtes doivent être calculées pour tous les couples de primitives dans la scène. Autrement dit la complexité du calcul du graphe de visibilité est $O(n^2)$ où n est la somme des arêtes et des polygones dans la scène.

La suite de cette section présente tout d'abord la solution mise en place pour unifier les trois types de requête de visibilité. Puis nous expliquons l'architecture développée pour limiter le coût du calcul complet d'un graphe de visibilité.

4.2.1 Requête de visibilité unifiée

Comme indiqué précédemment, le calcul du graphe de visibilité nécessite de prouver la visibilité de deux arêtes, ou d'une arête et d'un polygone, ou encore de deux polygones. Si l'on raisonne en terme d'ensemble de droites, cela correspond respectivement à des problèmes de dimensions 2, 3 et 4. C'est bien la preuve de la visibilité qui va être mémorisée par le graphe, et non une description complète. Pour deux polygones, une approche semblable à celle proposée par Haumont *et al.* [HMN05] (voir 1.4.11) serait donc suffisante. Il serait également possible de décliner cette technique pour prouver la visibilité de deux arêtes ou d'une arête et d'un polygone. Bien entendu l'espace de résolution serait à étudier. Plutôt que d'envisager une solution basée sur trois requêtes différentes à appliquer en fonction des trois types de couples de primitives, nous proposons d'unifier la résolution de ces trois problèmes.

Il s'agit de montrer que toute l'information dont nous avons besoin est contenue dans les bords des polytopes représentatifs de la visibilité de deux polygones. De manière générale, notre solution consiste

¹Le phénomène de réfraction est seulement pris en compte pour des scènes d'intérieur

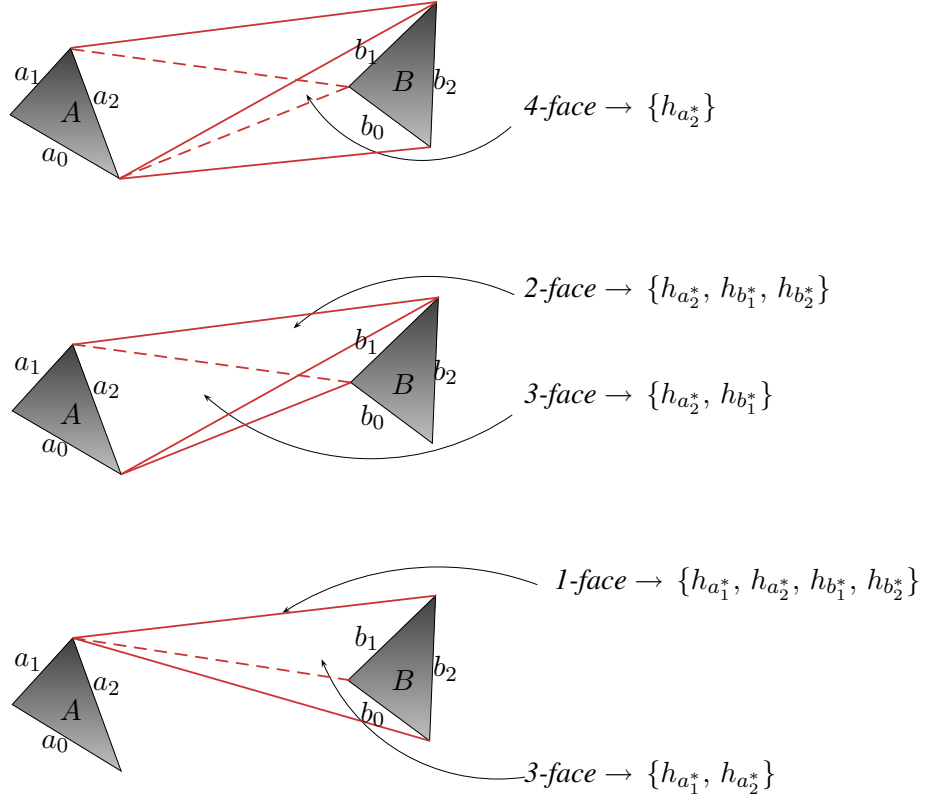


FIG. 4.2 – Illustration de la correspondance entre la visibilité des bords des polygones A et B , et les droites représentées par les k -faces du bord de \mathcal{P}_{AB} . Pour chaque exemple est donnée la description combinatoire de la k -face. En haut : les droites incidentes à une arête forment une 4-face de \mathcal{P}_{AB} . Au centre : les droites incidentes à une arête de A et à une arête de B correspondent à une 3-face de \mathcal{P}_{AB} . Celles incidentes à un sommet et une arête forment une 2-face. En bas : les droites incidentes à un sommet de A ou de B forment une 3-face du bord de \mathcal{P}_{AB} , celles incidentes à un sommet de A et un sommet de B , i.e. les *esl*, correspondent à une arête (1-face) de \mathcal{P}_{AB} .

à calculer dans l'espace de Plücker l'arbre VBSP représentatif de la visibilité de deux polygones A et B . Comme pour le précédent chapitre, cette étape fait référence aux travaux présentés dans le chapitre 2. Puis, si A et B sont visibles, un algorithme examine les k -faces (ou k -polytopes, voir la section 1.2.3) des polytopes du complexe sous-jacent à l'arbre VBSP², afin d'en extraire les visibilités relatives aux arêtes a_i de A et b_j de B . Ainsi l'information contenue dans le complexe de polytopes permet d'utiliser une seule requête, appliquée seulement aux couples de polygones de la scène. Nous expliquons dans la suite de cette section comment obtenir la visibilité des arêtes de deux polygones A et B via le complexe de polytopes représentatif de leur visibilité.

Comme indiqué dans la section 1.3.3 du chapitre 1, les droites critiques représentées par les bords d'un polytope ne sont pas nécessairement des droites tangentes aux arêtes de la géométrie dans \mathbb{R}^3 . Elles sont tangentes aux droites supportant ces arêtes. Cette distinction entre une arête et sa droite support n'a pas lieu d'être s'il s'agit du bord d'un polytope P caractérisant l'ensemble des droites passant à travers une séquence de polygones convexes. C'est donc en particulier le cas pour un polytope \mathcal{P}_{AB} représentant les droites intersectant deux polygones A et B .

²Par complexe de polytopes sous-jacent, nous entendons l'ensemble des polytopes contenus dans chaque feuille *visible* d'un arbre VBSP

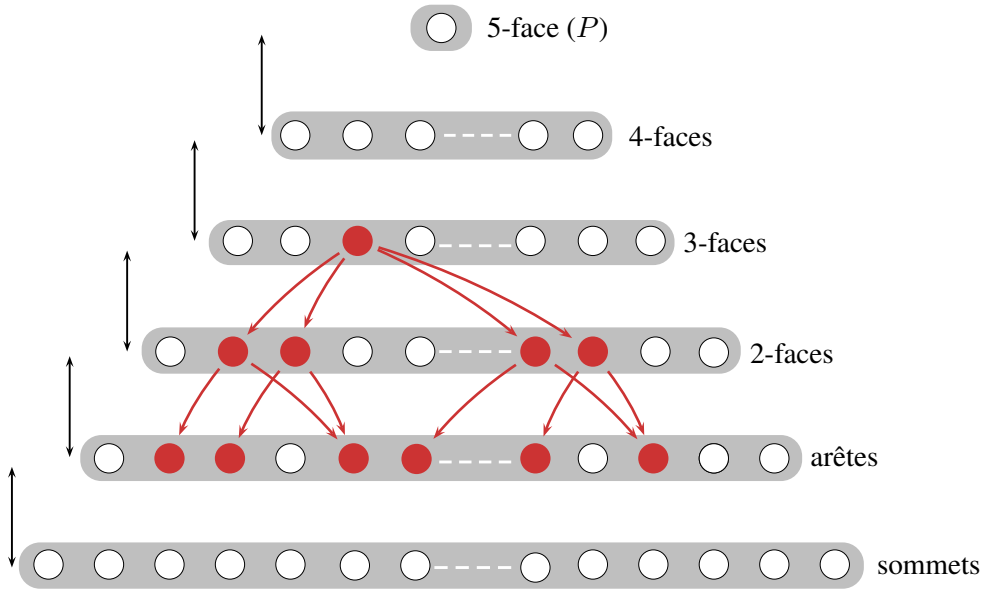


FIG. 4.3 – Cette figure illustre le graphe d’incidence (voir 1.2.3) d’un polytope P dans \mathbb{R}^5 (la 5-face représente P lui-même). Pour ne pas surcharger le schéma, les arcs entre toutes les k -faces ne sont pas représentés. Chaque k -face de P est un k -polytope dont le graphe d’incidence est contenu dans le graphe d’incidence de P : il s’agit du sous-graphe enraciné en la k -face. Ici, l’exemple est donné pour une 3-face. L’accès aux arêtes contenues dans une k -face se fait donc directement via le sous-graphe d’incidence de la k -face.

Dans la section 2.3.1 du chapitre 2, nous avons vu que la structure de \mathcal{P}_{AB} était prédictible et que son graphe d’incidence ne dépend que du nombre d’arêtes de A et de B . Il en résulte que la description combinatoire de chaque k -face du bord de \mathcal{P}_{AB} est également prédictible.

Notons $h_{e_i^*}$ la représentation par hyperplan dans l’espace de Plücker de la droite support à l’arête e_i d’un polygone. Puisque seules les droites réelles nous intéressent, nous limitons notre analyse aux k -faces contenant une arête intersectant la quadrique de Plücker. Par construction ces arêtes de \mathcal{P}_{AB} sont connues. La description combinatoire de chacune est $\{h_{a_i}, h_{a_{i+1}}, h_{b_j}, h_{b_{j+1}}\}$ et leur intersection avec la quadrique de Plücker est une *esl* passant par un sommet $s_{i,i+1}$ de A (sommet incident aux arêtes d’indice i et d’indice $i+1$) et un sommet $s_{j,j+1}$ de B . De ce point, nous pouvons énumérer toutes les k -faces de \mathcal{P}_{AB} qui représentent des droites réelles. Le tableau 4.1 présente leur description combinatoire.

A présent, faisons la correspondance entre les droites critiques incidentes aux arêtes de A et/ou de B et les k -faces du tableau 4.1 :

- Les droites critiques incidentes soit à une arête a_i , soit à une arête b_j correspondent à une 4-face du bord de \mathcal{P}_{AB} dont la description combinatoire est soit $\{h_{a_i^*}\}$, soit $\{h_{b_j^*}\}$.
- Les droites critiques incidentes à deux arêtes e_k et $e_{k'}$ appartenant à A et/ou à B correspondent à une 3-face du bord de \mathcal{P}_{AB} dont la description combinatoire est $\{h_{e_k^*}, h_{e_{k'}^*}\}$.
En particulier, si $e_k = a_i$ et $e_{k'} = b_j$, la 3-face est représentative des visibilités entre deux arêtes de A et B . Si $e_k = a_i$ et $e_{k'} = a_{i+1}$ (resp. $e_k = b_j$ et $e_{k'} = b_{j+1}$), la 3-face représente la visibilité de B (resp. A) depuis le sommet de A (resp. B) incident aux arêtes a_i et a_{i+1} (resp. b_j et b_{j+1}).
- Les droites critiques incidentes à deux arêtes (a_i, a_{i+1}) de A et à une arête b_j de B correspondent à une 2-face du bord de \mathcal{P}_{AB} dont la description combinatoire est $\{h_{a_i^*}, h_{a_{i+1}^*}, h_{b_j^*}\}$. Elle représente

Algorithme 4 : Pseudo-code déterminant les visibilitées des arêtes de deux polygones A et B après le calcul dans l'espace de Plücker du complexe de polytopes associé à la visibilité de A et B .

```

27 fonction contient_visibilites (Liste  $kfaces$ ) retourne booléen
28 Node  $face$ 
29 début
30   si  $kfaces = \emptyset$  alors
31     retourner faux
32   sinon
33     pour chaque  $face f$  de  $kfaces$  faire
34       si  $f$  intersecte la quadrique de Plücker alors
35         retourner vrai
36   retourner faux
37 fin

38 procedure visibilite_aretes (Polygone  $A, B$ , VBSP  $arbre$ )
39 Liste  $kfaces$ 
40 début
41   pour chaque arête  $a_i$  de  $A$  faire
42      $kfaces \leftarrow 4faces\_associees(a_i, arbre)$ 
43     si contient_visibilites( $kfaces$ ) alors
44        $marquer\_visibles(a_i, B)$ 
45   pour chaque arête  $b_j$  de  $B$  faire
46      $kfaces \leftarrow 4faces\_associees(b_j, arbre)$ 
47     si contient_visibilites( $kfaces$ ) alors
48        $marquer\_visibles(b_j, A)$ 
49   pour chaque arête  $a_i$  de  $A$  faire
50     pour chaque arête  $b_j$  de  $B$  faire
51        $kfaces \leftarrow 3faces\_associees(a_i, b_j, arbre)$ 
52       si contient_visibilites( $kfaces$ ) alors
53          $marquer\_visibles(a_i, b_j)$ 
54 fin

55 procedure requete_unifiee (Polygone  $A, B$ )
56 VBSP  $arbre$ 
57 début
58    $arbre \leftarrow VisibilitePoly2Poly(A, B)$ 
59   si  $arbre.polytopes \neq \emptyset$  alors
60      $marquer\_visibles(A, B)$ 
61      $visibilite\_aretes(A, B, arbre)$ 
62 fin

```

donc la visibilité d'un sommet de A avec une arête de B . Il en est de même pour un couple d'arêtes de B et une arête de A .

- Les esl incidentes à un sommet de A et à un sommet de B correspondent à une arête (1-face) du bord de \mathcal{P}_{AB} dont la description combinatoire est $\{h_{a_i^*}, h_{a_{i+1}^*}, h_{b_j^*}, h_{b_{j+1}^*}\}$.

La figure 4.2 illustre ces ensembles de droites critiques en donnant la k -face associée dans le bord de \mathcal{P}_{AB} . Dans notre contexte, seule la visibilité des arêtes de A et de B est nécessaire. Par conséquent, les

| k -face | descriptions combinatoires |
|-----------|--|
| arête | $\{h_{a_i}, h_{a_{i+1}}, h_{b_j}, h_{b_{j+1}}\}$ |
| 2-face | $\{h_{a_i}, h_{a_{i+1}}, h_{b_j}\}$ ou $\{h_{a_i}, h_{b_j}, h_{b_{j+1}}\}$ |
| 3-face | $\{h_{a_i}, h_{a_{i+1}}\}$ ou $\{h_{b_j}, h_{b_{j+1}}\}$ ou $\{h_{a_i}, h_{b_j}\}$ |
| 4-face | $\{h_{a_i}\}$ ou $\{h_{b_j}\}$ |

TAB. 4.1 – Ce tableau résume les descriptions combinatoires des k -faces constitutives du bord d'un polytope \mathcal{P}_{AB} . Plus précisément, il s'agit des k -faces possédant une intersection avec la quadrique de Plücker.

k -faces du bord de \mathcal{P}_{AB} concernées sont les 4-faces et les 3-faces dont les descriptions combinatoires correspondent à une arête ou à un couple d'arêtes de A et B . Suite au calcul de l'arbre VBSP associé aux deux polygones, s'il demeure dans le complexe de polytopes sous-jacent tout ou partie de ces 4-faces ou 3-faces, cela signifie la visibilité des arêtes concernées. Si tel est le cas, il est cependant possible que la k -face soit disjointe de la quadrique de Plücker. Aussi l'intersection avec cette dernière est à vérifier. Pour résumer, les k -faces à rechercher sont les suivantes :

- Les 4-faces de description $\{h_{a_i^*}\}$ (resp. $\{h_{b_j^*}\}$), possédant une intersection non vide avec la quadrique de Plücker, représentatives des visibilités entre l'arête a_i (resp. b_j) et le polygone B (resp. A).
- Les 3-faces de description combinatoire $\{h_{a_i^*}, h_{b_j^*}\}$, possédant une intersection non vide avec la quadrique de Plücker, représentatives des visibilités entre les arêtes a_i et b_j .

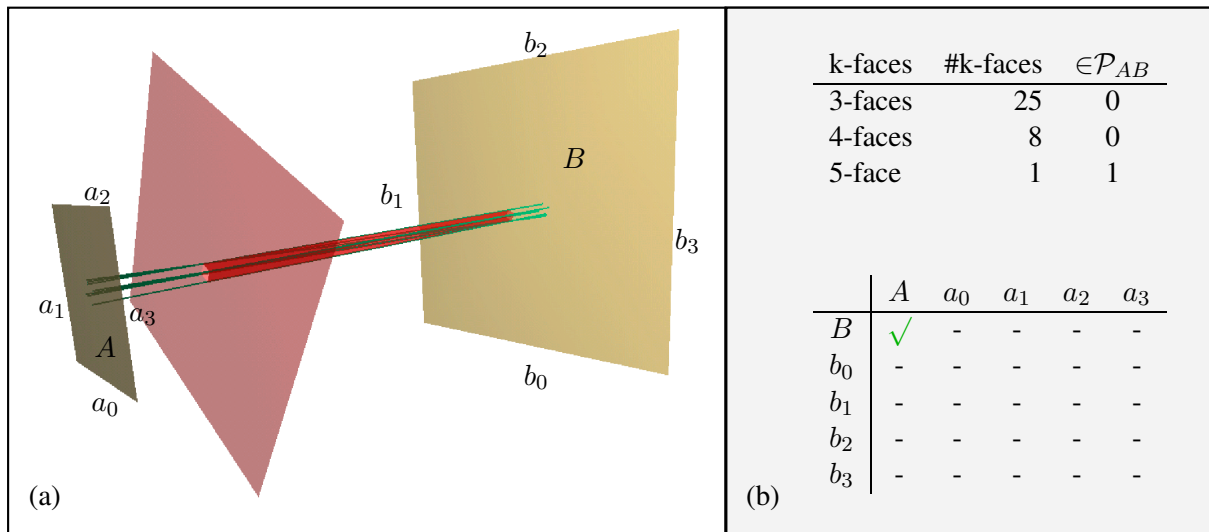


FIG. 4.4 – (a) Un polytope représente l'ensemble des visibilités entre le polygone A et le polygone B . Il s'agit ici des droites passant dans la section de tube ; il n'existe pas de visibilité avec les arêtes a_i de A ou b_j de B . La capture d'écran montre les *esl* du polytope en question. (b) En haut : le polytope possède vingt-cinq 3-faces et huit 4-faces qui ne sont pas identifiées par l'algorithme comme appartenant initialement au bord de \mathcal{P}_{AB} . La 5-face, *i.e.* le polytope lui-même, correspond donc à un ensemble de droites strictement incluses dans \mathcal{P}_{AB} . En bas : le tableau donne l'interprétation de ce résultat en terme de visibilité (arête, arête), (arête, polygone) et (polygone, polygone).

Parcourir toutes les 4 et 3-faces de tous les polytopes en vérifiant pour chacune sa description combinatoire puis son intersection avec la quadrique de Plücker est une solution valide mais peu efficace. Pour

restreindre le nombre de k -faces à examiner, les 4-faces et les 3-faces du polytope \mathcal{P}_{AB} sont initialement marquées de manière à identifier celles qui correspondent à la visibilité des arêtes. Rappelons que nous disposons du graphe d'incidence complet de \mathcal{P}_{AB} . Cette étape se résume donc à un simple parcours des nœuds du graphe d'incidence décrivant l'ensemble des 4-faces et des 3-faces de \mathcal{P}_{AB} . La description combinatoire de chacune indique directement la correspondance avec une arête a_i ou b_j , ou un couple d'arête (a_i, b_j) . Pendant le calcul de l'arbre VBSP, l'intersection d'un polytope par un hyperplan conduit

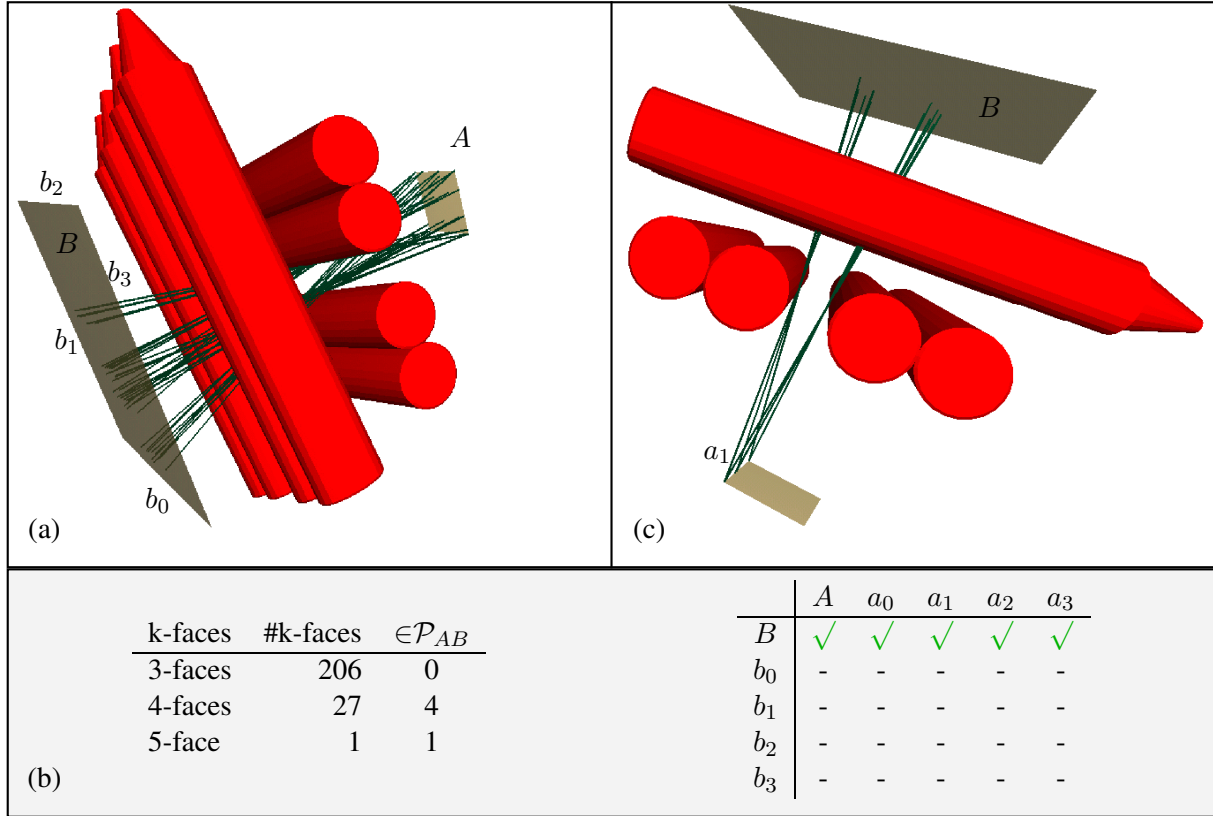


FIG. 4.5 – (a) les *esl* du polytope représentant les visibilitées entre A et B . Ici, aucune arête b_j de B n'est visible, mais A et ses arêtes a_i sont visibles depuis B . (c) A gauche : aucune des 206 3-faces du polytope ne possède une description combinatoire de type $\{h_{a_i^*}, h_{b_j^*}\}$, les arêtes b_j étant masquées. Quatre des vingt-sept 4-faces sont éléments de \mathcal{P}_{AB} et de description combinatoire $\{h_{a_i^*}\}$. Elles représentent les visibilitées de chaque arête a_i avec le polygone B . A droite : le tableau résume les visibilitées identifiées. (c) A titre d'exemple, la capture d'écran représente les visibilitées de l'arête a_1 via les *esl* de la 4-face de description combinatoire $\{h_{a_1^*}\}$.

à l'intersection successive de ses k -faces. Lors de cette étape, chaque k -face intersectée f donne lieu à deux k -faces f^+ et f^- classées de part et d'autre de l'hyperplan. Si f est une 4-face ou 3-face marquée, alors f^+ et f^- héritent de cette marque. Ainsi, pour chaque a_i et b_j et pour chaque couple d'arêtes (a_i, b_j) , la liste des 4-faces et 3-faces qui leur sont associées est implicitement maintenue pendant le calcul de l'arbre VBSP.

L'algorithme 4 donne le pseudo-code de la requête *unifiée* (ligne 29) pour le calcul de la visibilité de deux polygones A et B et de leurs arêtes a_i et b_j . L'arbre VBSP pour A et B est préalablement calculé (ligne 32). Si l'arbre est représentatif de visibilitées (ligne 33), les deux polygones sont définis visibles (ligne 34) et la visibilité de leurs arêtes est extraite (ligne 35). Pour chaque arête a_i de A (ligne 15), on accède directement à la liste des 4-faces à vérifier (ligne 16) grâce au marquage effectué sur les bords

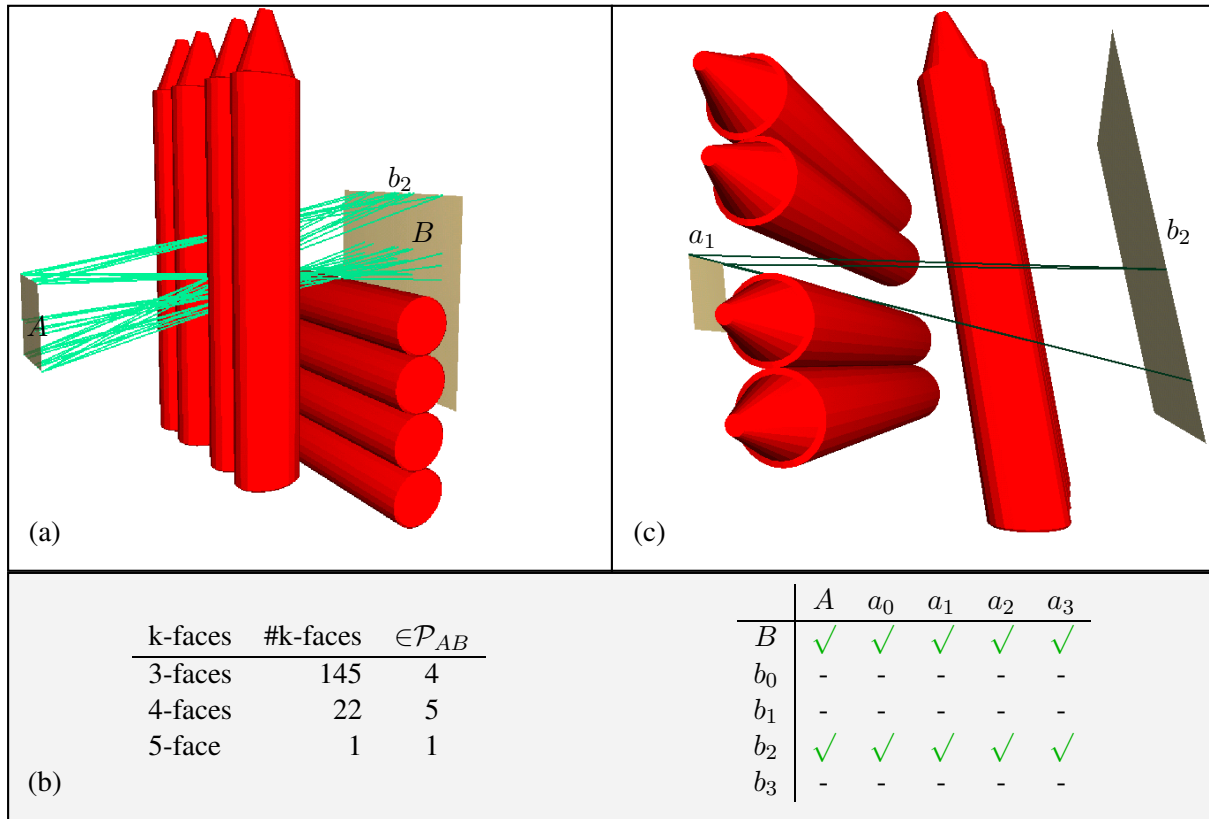


FIG. 4.6 – (a) Un polytope, visualisé par ses *esl*, représente l'ensemble des visibilitées entre A et B . (b) A gauche : il possède 145 3-faces et vingt-deux 4-faces. L'algorithme a identifié quatre 3-faces appartenant à \mathcal{P}_{AB} , et dont les descriptions combinatoires indiquent qu'elles correspondent à la visibilité des arêtes (a_i , b_2). Cinq 4-faces sont identifiées comme élément de \mathcal{P}_{AB} . Quatre d'entre elles représentent la visibilité des arêtes a_i et de B , la cinquième celle de b_2 et de A . A droite : le tableau résume les visibilitées identifiées. (c) A titre d'exemple, la capture d'écran montre l'ensemble des visibilitées des arêtes (a_1 , b_2), *i.e.* les *esl* de la 3-faces de description combinatoire $\{h_{a_1^*}, h_{b_2^*}\}$.

de \mathcal{P}_{AB} , et maintenu pendant le calcul de l'arbre VBSP. Si cette liste est vide, aucune visibilité n'existe depuis a_i (lignes 4-5). Sinon il faut vérifier qu'au moins une 4-face possède une intersection avec la quadrique de Plücker (lignes 6-9), auquel cas l'arête a_i est définie visible avec B (ligne 18). Le procédé est appliqué aux arêtes b_j de B (lignes 19-22) et aux couples d'arêtes (a_i , b_j) (lignes 23-27). Comme l'illustre la figure 4.3 le graphe d'incidence des faces d'un polytope donne directement accès aux arêtes contenues dans n'importe quelle k-face de son bord. Dès lors, le test d'intersection d'une k-face avec la quadrique de Plücker est immédiat.

Les figures 4.4, 4.6 et 4.5 donnent quelques exemples d'extraction de la visibilité d'arêtes. La figure 4.4 présente une configuration où aucune arête de A ou B n'est visible. Sur la figure 4.6, seules les arêtes de A sont visibles. Enfin sur la figure 4.5, seules les arêtes de A et une arête de B sont visibles.

La méthode que nous venons de présenter permet, à partir du complexe de polytopes de deux polygones A et B , de déterminer les visibilitées de leurs arêtes. De cette manière, le calcul du graphe de visibilité peut être ramené à des requêtes polygones à polygones, sans devoir prouver séparément la visibilité de toutes les combinaisons arête-arête et arête-polygone. On peut noter que cette méthode s'étend naturel-

lement à la visibilité des sommets de A et B en étendant le marquage initial du bord de \mathcal{P}_{AB} à ses autres k -faces ($k > 0$).

4.2.2 Graphe de visibilité : calcul hiérarchique mixte

Cette section présente le calcul du graphe de visibilité pour un environnement donné. Plus précisément, un nœud de ce graphe correspond à un polygone, ou bien à un dièdre (voir figure 4.7). Les polygones sont orientés, et un dièdre est défini par deux polygones adjacents par une arête. De fait, si l'on détermine une relation de visibilité pour une arête donnée, c'est à son dièdre dans le graphe qu'un arc est ajouté.

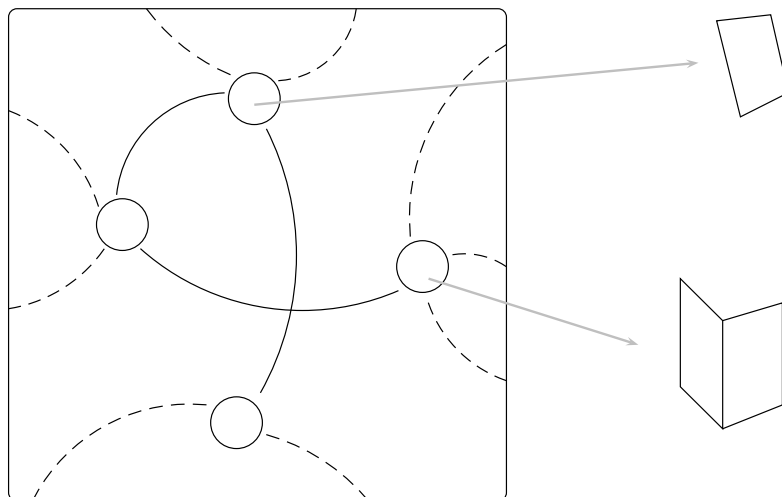


FIG. 4.7 – Aperçu d'une partie d'un graphe de visibilité : un nœud du graphe correspond à un polygone ou un dièdre de la scène. Deux nœuds sont reliés par un arc si les primitives associées sont visibles.

L'approche présentée tend à limiter le temps de pré-calcul du graphe. En effet, pour une scène comprenant n polygones, la complexité du calcul du graphe est $O(n^2)$ puisqu'il nécessite d'effectuer $\frac{n(n-1)}{2}$ requêtes de visibilité. Le graphe ne devant être calculé qu'une seule fois par scène, le temps de pré-calcul n'est pas un facteur de première importance. Cependant, même si la taille des scènes³ utilisées en propagation excède très rarement 100 000 polygones, le milliard de requêtes à effectuer est rapidement atteint et dépassé. Pour conserver des temps de pré-calcul praticables, et dans l'optique d'un accroissement de la taille des scènes (si l'on monte en fréquence par exemple), il est nécessaire de casser cette complexité. Nous donnons tout d'abord une vue d'ensemble de notre approche.

Vue d'ensemble

Le calcul du graphe est d'une part mixte, au sens où il combine preuve et description de la visibilité, d'autre part hiérarchisé. Initialement, le graphe ne contient que les nœuds associés à chaque primitive de la scène : aucune relation de visibilité n'existe. Une hiérarchie de boîtes englobantes \mathcal{H} est utilisée pour organiser la scène. Pour chaque feuille f de \mathcal{H} , sa visibilité est hiérarchiquement testée avec le reste de la géométrie. Pour prouver la visibilité de f avec une région de \mathcal{H} (feuilles ou nœuds internes), une requête adaptée des travaux de Haumont *et al.* [HMN05] est utilisée. Lorsque deux feuilles sont déterminées visibles, alors la visibilité des polygones et des arêtes qu'elles contiennent est calculée en utilisant la requête unifiée présentée dans la section précédente (voir l'algorithme 4). Le graphe de visibilité est mis

³En propagation, une très grande scène comporte 50 000 polygones environ

à jour en conséquence.

Les différents aspects de cette approche sont à présent justifiés.

Le choix d'une hiérarchie de boîtes englobantes n'est pas limitant, et d'autres structures pourraient convenir. Cependant, la hiérarchie de boîtes englobantes permet d'avoir des régions "au plus près" de la géométrie contenue, et évite de devoir gérer la possibilité qu'un polygone appartienne à plusieurs régions. On objectera qu'une méthode telle que [HMN05] nécessite de lancer des rayons dans la scène. Pour cette tâche, la hiérarchie de boîtes englobantes est reconnue comme n'étant pas la meilleure des structures accélératrices [HPP00], voire comme étant la moins efficace. Pour ce, un *kd-tree* est donc utilisé comme structure annexe dédiée à cette tâche.

S'il est nécessaire de calculer l'arbre VBSP de deux polygones pour extraire la visibilité de leurs arêtes, cette information est superflue s'il s'agit de la visibilité de deux régions de la hiérarchie \mathcal{H} . Dans ce dernier cas, seule la preuve de leur visibilité compte. Par conséquent, il est plus efficace d'utiliser une requête spécifique tant que la géométrie des feuilles n'est pas atteinte. La question qui se pose est : quelle preuve de visibilité utiliser ? Et de quelle nature : exacte ou conservative ? Nous aurions pu envisager l'utilisation de techniques conservatives conçues pour le calcul de PVS dans des applications de type "ballade virtuelle" [COCD03]. Cette possibilité a été écartée pour deux raisons :

- Pour que notre approche demeure évolutive, elle doit pouvoir facilement s'adapter de la 2.5D à la 3D. Les méthodes conservatives utilisées pour les ballades virtuelles sont dédiées à la nature 2.5D des environnements utilisés. Soit leur transposition à la 3D est tout simplement impossible, soit elle ne pourrait être faite sans altérer l'efficacité de la méthode.
- Le contexte de ballade virtuelle est particulier : il implique une forte occultation, car la hauteur et la taille d'une cellule de vue sont en général très inférieures à celles des bâtiments environnants. Dans notre cas, s'agissant de la visibilité de deux bâtiments, ces valeurs sont voisines, et l'importance de l'occultation très variable. Dans ces conditions, la proportion de géométrie définie conservativement visible à tort deviendrait rapidement conséquente.

Face au besoin de disposer d'une requête précise, le choix d'une méthode exacte s'impose. Bittner *et al.* [BWW05] ont récemment présenté un calcul exact de PVS dédié aux environnements urbains. Il s'agit d'une évolution des travaux présentés dans la thèse de Bittner [Bit02]. Leur solution calcule dans le plan 2D du sol la visibilité entre l'empreinte d'une cellule de vue, et l'empreinte des bâtiments. La version 2D des arbres d'occultation est alors utilisée. Le traitement de la dimension verticale est ramené au calcul de l'existence d'au moins une visibilité à travers une séquence de portails [TH91, Tel92b] (voir la section 1.4.4).

Ce type d'approche n'est malheureusement pas généralisable à des environnements 3D. Pour cette raison, nous avons partiellement repris l'algorithme proposé par Haumont *et al.* [HMN05]. Plus précisément, c'est le principe de lancer un rayon choisi aléatoirement dans un polytope qui est repris (voir la section 1.4.11 du chapitre 1). En revanche, l'extraction de la silhouette des objets n'est pas exploitée : les bâtiments d'une scène ne sont pas des objets suffisamment détaillés pour justifier une approche basée silhouette. De plus, les algorithmes utilisés pour effectuer les opérations CSG dans l'espace de Plücker demeurent identiques à ceux mentionnés dans le chapitre 2.

L'algorithme 5 donne le pseudo-code de cette version simplifiée de [HMN05]. L'algorithme récursif est initialisé avec le polytope représentatif de la visibilité de deux polygones A et B (ligne 67). Une droite est choisie dans le polytope (ligne 44) en considérant l'intersection avec la quadrique de Plücker d'un segment dont les sommets sont choisis aléatoirement comme combinaisons linéaires de la V-représentation du polytope. Cette droite est utilisée pour construire un rayon (ligne 45) lancé dans la scène depuis le polygone A (ligne 46). Si le premier objet intersecté par le rayon est le polygone B (ligne 47), A et B

Algorithme 5 : Version simplifiée de l'algorithme proposé par Haumont [HMN05] pour prouver la visibilité de deux polygones. Cet algorithme est adapté de celui présenté dans [HMN05] mais il n'utilise pas la silhouette des objets.

```

63 fonction preuve_recursive (Polytope  $P$ , Polygone  $A, B$ ) retourne booléen
64 Droite  $l$ 
65 Rayon  $\text{rayon\_AversB}$ 
66 Polytope  $P^+$ 
67 Polygone  $\text{bloqueur}$ 
68 Plucker  $hplan$ 
69 début
70    $l \leftarrow \text{droite\_aleatoire}(P)$ 
71    $\text{rayon\_AversB} \leftarrow \text{creer\_rayon}(A, l)$ 
72    $\text{bloqueur} \leftarrow \text{lancer\_rayon}(\text{rayon\_AversB})$ 
73   si  $\text{bloqueur} = B$  alors
74     | retourner vrai
75   sinon
76     pour chaque  $\text{arete } e_i \text{ de bloqueur faire}$ 
77       |  $hplan \leftarrow \text{map2Plucker}(e_i)$ 
78       | si  $hplan \cap P \neq \emptyset$  alors
79         |  $P^+ \leftarrow hplan^+ \cap P$ 
80         |  $P \leftarrow hplan^- \cap P$ 
81         | si  $\text{preuve\_recursive}(P^+, A, B)$  alors
82           | retourner vrai
83         | fin
84       | fin
85     | fin
86   fin
87    $\text{destruire\_polytope}(P)$ 
88   retourner faux
89 fin

90 fonction preuve_visibilite (Polygone  $A, B$ ) retourne booléen
91 Polytope  $P$ 
92 début
93    $P \leftarrow \text{creer\_polytope}(A, B)$ 
94   retourner  $\text{preuve\_recursive}(P, A, B)$ 
95 fin

```

sont visibles et la requête se termine (ligne 48). Sinon, l'objet retourné est un bloqueur. Chacune de ses arêtes est convertie en hyperplan dans l'espace de Plücker (ligne 51). Si celui-ci intersecte le polytope (ligne 52-54), l'algorithme est rappelé sur la partie du polytope représentant un ensemble de droites qui assurément "évite" le bloqueur (ligne 55). Si une visibilité est trouvée dans cette partie, A et B sont visibles et le procédé se termine (ligne 56). Dans le cas contraire, lorsque toutes les arêtes du bloqueur ont été traitées, la partie du polytope qui demeure est représentative des droites intersectées par le bloqueur. Celui-ci est donc éliminé (ligne 61) et toutes les droites initialement contenues dans le polytope sont occultées (ligne 62).

Revenons à présent plus en détail sur l'ensemble du procédé à travers les pseudo-codes des requêtes de visibilités utilisées, et du calcul complet d'un graphe de visibilité d'un environnement 2.5D.

Algorithme

Algorithme 6 : Pseudo-code des deux requêtes de visibilité utilisées pour le calcul hiérarchique des relations de visibilité dans une scène 2.5D. Ligne 70 : la fonction `regions_visibles` détermine la visibilité de deux régions en testant la visibilité des boîtes englobantes qui les délimitent. Exploitant la nature 2.5D des scènes, seule la visibilité des faces supérieures des boîtes est testée. Ligne 81 : la procédure `visibilite_primitives` détermine la visibilité des polygones et des arêtes contenues dans deux régions distinctes en utilisant la procédure `requete_unifiee` présentée précédemment par l’algorithme 4.

```

96 fonction regions_visibles (Feuille  $f$ , Nœud  $n$ ) retourne booléen
97   Polygone  $A, B$ 
98   début
99     si  $f.bbox \cap n.bbox \neq \emptyset$  alors
100       | retourner vrai
101     sinon
102       |  $A \leftarrow \text{face\_superieure}(f.bbox)$ 
103       |  $B \leftarrow \text{face\_superieure}(n.bbox)$ 
104       | retourner preuve_visibilite( $A, B$ )
105     fin
106   fin

107 procedure visibilite_primitives (Feuille  $f_1$ , Feuille  $f_2$ )
108   début
109     pour chaque polygone  $p_i$  de  $f_1$  faire
110       | pour chaque polygone  $p_j$  de  $f_2$  faire
111         | | requete_unifiee( $p_i, p_j$ )
112       | fin
113     fin
114   fin

```

L’algorithme 6 donne le pseudo-code des deux requêtes de visibilité utilisées. Considérant la hiérarchie de boîtes englobantes \mathcal{H} , deux nœuds de cette hiérarchie correspondent à deux régions. Tant que l’algorithme teste la visibilité de régions de \mathcal{H} , les calculs de visibilité sont effectués en mode *preuve*, pour démontrer efficacement la visibilité ou l’invisibilité de deux régions. La requête alors utilisée est donnée ligne 70. Elle vérifie que les régions sont disjointes (ligne 73), auquel cas les faces supérieures de chaque région sont isolées (lignes 76-77). La visibilité des deux régions est établie en testant uniquement la visibilité de leurs faces supérieures (ligne 78) en utilisant l’approche de Haumont *et al.* La nature 2.5D des environnements traités autorise cette simplification. On notera cependant que le procédé s’étend naturellement à des environnements 3D en testant toutes les faces délimitant une région.

Dès que la géométrie des feuilles est atteinte, les calculs de visibilité basculent en mode *unifié*, afin de calculer le complexe de polytopes complet et d’en extraire toutes les informations utiles à la mise à jour du graphe de visibilité. La requête alors utilisée est donnée ligne 81 et consiste simplement à appliquer la requête unifiée précédemment présentée (voir l’algorithme 4) à chaque couple de polygones contenus (lignes 83-85).

L’algorithme 7 donne le pseudo-code du calcul de toutes les relations de visibilité entre les arêtes et les polygones d’une scène. La procédure présentée à la ligne 103 est appliquée à la hiérarchie de boîtes englobantes \mathcal{H} . Elle est récursivement appelée sur les fils de chaque nœud interne (lignes 113-114). Le nœud courant est marqué (ligne 115) afin de signifier que la visibilité de la géométrie contenue dans la région représentée est déjà testée. Lorsqu’une feuille est atteinte (ligne 105), la visibilité entre les arêtes

Algorithme 7 : Pseudo-code pour le calcul hiérarchiquement de l'ensemble des relations de visibilité entre toutes les arêtes et tous les polygones de la scène. Ligne 89 : le procédé s'appuie sur une subdivision hiérarchique spatiale de la scène, ici une hiérarchie de boîte englobante. Ligne 90 : `visibilite_feuille` calcule hiérarchique la visibilité de la géométrie contenue dans une feuille avec le reste de la scène. Ligne 103 : `visibilite_environnement` applique récursivement `visibilite_feuille` à chaque feuille de la hiérarchie \mathcal{H} , et marque chaque région traitée pour qu'elle soit ignorée par la suite.

```

115 bbox_hierarchie  $\mathcal{H}$ 
116 procedure visibilite_feuille (Feuille  $f$ , Nœud  $n$ )
117 début
118   si  $n.visite = \text{faux}$  alors
119     si regions_visibles ( $f, n$ ) alors
120       si  $n.type = \text{feuille}$  alors
121         | visibilite_primitives ( $f, n$ )
122       sinon
123         | visibilite_feuille ( $f, n.filsg$ )
124         | visibilite_feuille ( $f, n.filsd$ )
125       fin
126     fin
127   fin
128 fin

129 procedure visibilite_environnement (Nœud  $n$ )
130 début
131   si  $n.type = \text{feuille}$  alors
132     pour chaque polygone  $p_i$  de  $n$  faire
133       pour chaque polygone  $p_j, j > i$  de  $n$  faire
134         | requete_unifiee ( $p_i, p_j$ )
135       fin
136     fin
137     visibilite_feuille ( $n, \mathcal{H}$ )
138   sinon
139     visibilite_environnement ( $n.filsg$ )
140     visibilite_environnement ( $n.filsd$ )
141      $n.visite \leftarrow \text{vrai}$ 
142   fin
143 fin

```

et les polygones contenus dans la feuille est tout d'abord calculée (lignes 106-110). Puis la visibilité avec le reste de la hiérarchie est testée (ligne 11).

Cette procédure ne s'applique qu'aux régions de \mathcal{H} non encore testées (ligne 92). Elle commence par prouver (algorithme 6, ligne 70) la visibilité de la feuille avec le nœud courant de la hiérarchie (ligne 93). S'il s'agit de deux feuilles (ligne 94) alors le calcul (algorithme 6, ligne 81) des relations de visibilité entre toutes les primitives contenues est effectué (ligne 95). S'il s'agit d'un nœud interne, le procédé est répété récursivement sur ses fils (lignes 97-98).

Les pseudo-codes de ces algorithmes sont volontairement simplifiés pour ne pas nuire à la compréhension de l'ensemble du procédé. Par conséquent, ils ne font pas apparaître une optimisation adaptée de l'architecture logicielle proposée par Nirenstein [NBG02] : les bloqueurs virtuels. Dans notre contexte, il s'agit pour la procédure `visibilite_feuille` de marquer les régions qu'elle détermine invisibles pour une feuille f donnée. Ces régions invisibles sont ensuite utilisées comme bloqueurs virtuels lors du calcul

d'un arbre VBSP entre un polygone contenu dans f et un polygone contenu dans une feuille visible depuis f . Rappelons que les bloqueurs de deux polygones A et B sont sélectionnés en considérant la géométrie contenue dans l'enveloppe convexe de A et B . Cette sélection est effectuée hiérarchiquement sur \mathcal{H} . Si une région invisible s'avère posséder une intersection avec l'enveloppe convexe de A et de B , les faces de sa boîte englobante sont directement retournées comme bloqueurs.

4.2.3 Résultats

Cette section présente les résultats des tests ⁴ effectués pour évaluer l'approche présentée. Deux scènes de tests sont utilisées, représentant une zone restreinte (scène A) et une zone plus étendue (scène B) d'un quartier de Paris. Deux séries de tests sont présentées :

- La première série porte sur le précalcul du graphe de visibilité. Elle indique les temps de précalcul total et distingue ceux consommés par chaque type de requête de visibilité. Elle présente également les statistiques sur les visibilité entre chaque type de nœud du graphe, *i.e.* un polygone ou un dièdre.
- La seconde série de tests montre l'impact du graphe de visibilité sur les temps de calculs de zones de couverture. Pour ce, le graphe de visibilité calculé est couplé à un simulateur de propagation des ondes électromagnétiques basé sur le principe des source-images. Il s'agit du logiciel CRT évoqué plus haut (voir 4.1.3).

Au préalable, précisons la méthode utilisée pour construire les hiérarchies de boîtes englobantes pour chaque scène. L'algorithme de construction est de type *top-down*, *i.e.* la boîte englobante de la scène entière est récursivement subdivisée jusqu'à vérification d'un critère d'arrêt. Un nœud de la hiérarchie possède deux fils. Par souci d'efficacité, chaque subdivision cherche à satisfaire les deux critères suivants :

- Les boîtes englobantes des deux fils d'un nœud sont disjointes. Dans le cas contraire, la visibilité de leur intersection sera testée pour chaque fils, ce qui est redondant.
- Le volume des boîtes englobantes des fils est minimal. Plus une région est petite, moins la visibilité depuis cette région est importante. L'idée ici est donc d'améliorer l'efficacité des requêtes de type preuve.

Pour ce faire, la subdivision d'un nœud s'appuie sur une heuristique qui cherche en premier lieu un plan n'intersectant aucun polygone contenu dans la région du nœud et, en second lieu, minimise le volume des boîtes englobantes des fils (c'est une heuristique couramment utilisée en détection de collision). Si aucun plan de subdivision n'est privilégié par ces heuristiques, l'algorithme utilise un plan médian par défaut.

Sur la scène de test A (resp. B), les critères d'arrêts utilisés sont : une profondeur maximale de 14 (resp. 16), ou un nombre de polygones inférieur ou égal à 3 (resp. 3) pour une région donnée. Ces valeurs tendent à ce qu'une feuille contienne partiellement un bâtiment.

Première série de tests : calcul du graphe de visibilité

Le graphe de visibilité pour la scène A possède 3 970 nœuds dont 2 163 polygones et 1 807 dièdres. Le temps nécessaire à son calcul est de 24 minutes et 20 secondes. Le tableau 4.2 montre le détail de ce calcul. Dans le cas de la scène B, le graphe de visibilité comporte 12 485 nœuds dont 6693 polygones et 5792 dièdres. Le temps nécessaire à son calcul et au traitement des couples de polygones distincts dans

⁴Tous les calculs ont été effectués sur un AMD64 3200+ (2.2 GHz) avec 3Go de mémoire

| | Nombre d'appels | Temps consommé | | Couples de polygones traités | |
|-------------------|-----------------|----------------|-------|------------------------------|-------|
| | | Durée | % | Nombre | % |
| Requêtes preuves | 139 788 | 4 min 19 s | 17.74 | 1 434 254 | 61.34 |
| Requêtes unifiées | 903 494 | 19 min 59 s | 82.12 | 903 949 | 38.66 |

TAB. 4.2 – Détails du calcul du graphe de visibilité sur la scène A. Première colonne : type de requête considérée. Seconde colonne : nombre d'appels pour chaque requête. Troisième colonne : temps consommé par chaque type de requêtes. Quatrième colonne : la même information que la troisième colonne mais en pourcentage du temps total de calcul du graphe. Cinquième colonne : le nombre de couples de polygones traités par chaque type de requête. Sixième colonne : la même information que la cinquième colonne, mais en pourcentage des 2 381 653 couples de polygones distincts dans la scène A.

la scène B est de 4 heures 14 minutes et 5 secondes. Comme pour la première scène, le tableau 4.3 donne les détails du calcul pour la scène B.

Avec 61.34 % des couples traités pour 17.74 % du temps total consommé pour la scène A, et 75.95 %

| | Nombre d'appels | Temps consommé | | Couples de polygones traités | |
|-------------------|-----------------|-----------------|-------|------------------------------|-------|
| | | Durée | % | Nombre | % |
| Requêtes preuves | 910 289 | 31 min 58 s | 12.58 | 17 008 961 | 75.95 |
| Requêtes unifiées | 5 385 817 | 3 h 42 min 03 s | 87.38 | 5 385 817 | 24.05 |

TAB. 4.3 – Détails du calcul du graphe de visibilité pour la scène B. Première colonne : type de requête considérée. Seconde colonne : nombre d'appels pour chaque requête. Troisième colonne : temps consommé par chaque type de requête. Quatrième colonne : la même information que la troisième colonne mais en pourcentage du temps total de calcul du graphe. Cinquième colonne : le nombre de couples de polygones traités par chaque type de requête. Sixième colonne : la même information que la cinquième colonne, mais en pourcentage des 22 394 778 couples de polygones distincts dans la scène B.

des couples traités pour 12.58 % du temps total consommé pour la scène B, les requêtes de type preuve s'avèrent très rentables. Cela ne signifie pas que ces requêtes sont nécessairement plus rapides. Le temps moyen d'une requête de type preuve est de $1.85 \cdot 10^{-3}$ seconde sur la scène A et $2.1 \cdot 10^{-3}$ seconde sur la scène B. En comparaison, une requête unifiée consomme en moyenne $1.33 \cdot 10^{-3}$ seconde sur la scène A et $2.47 \cdot 10^{-3}$ seconde sur la scène B. Ces temps sont logiques : en cas d'invisibilité de deux régions, une requête de type preuve est aussi coûteuse que le calcul "classique" du complexe de polytopes ; il conduit à la suppression totale de l'ensemble de droites initial par soustraction des différents bloqueurs. En outre, le coût du calcul est sensible aux dimensions des régions testées, qui sont importantes pour les premiers niveaux des hiérarchies de boîtes englobantes. L'intérêt des requêtes de type preuve n'est donc pas de déterminer rapidement des régions mutuellement invisibles, mais essentiellement de ne pas perdre de temps lorsqu'elles sont visibles.

Cependant, chaque fois que deux régions (*i.e.*, deux nœuds de la hiérarchie de boîtes englobantes) contenant respectivement n_1 et n_2 polygones sont prouvées comme étant invisibles, c'est $n_1 \times n_2$ couples de polygones qui sont éliminés du processus de calcul, d'où son efficacité. En comparaison, une requête unifiée élimine à chaque appel un seul couple de polygones (quel que soit le résultat de la requête). Le tableau 4.4 donne pour chaque scène les pourcentages de relations de visibilité établies entre chaque type de nœud d'un graphe. On constate un pourcentage de visibilité supérieur pour les dièdres. L'ensemble des visibilités d'un dièdre est au pire l'union des visibilités des deux polygones qui lui sont adjacents. De plus un dièdre correspond très souvent à un angle de rue, ou l'angle d'un toit avec un mur, qui sont

| Relation de visibilité | Graphe de visibilité | |
|------------------------------|----------------------|---------|
| | Paris A | Paris B |
| Visibilité polygone-polygone | 2.58 % | 1.67 % |
| Visibilité polygone-dièdre | 4.68 % | 2.97 % |
| Visibilité dièdre-dièdre | 6.40 % | 4.09 % |

TAB. 4.4 – Pourcentage de relations de visibilité entre chaque type d'élément d'un graphe pour les scènes A et B.

des positions offrant une visibilité accrue du reste de la scène. Ces configurations se retrouvent dans les pourcentages de visibilité calculés pour les dièdres.

Les valeurs présentées sont à mettre en correspondance avec la réduction du nombre de combinaisons à tester que permettra le graphe à chaque interaction. Considérons un exemple simple : la recherche combinatoire de tous les trajets entre un émetteur et un récepteur en 2 réflexions sur la scène A (2163 polygones). Dans le cas d'une approche *brute-force*, ce cas de figure nécessite de tester $2163^2 = 4\,678\,569$ combinaisons de polygones. En utilisant l'information contenue dans le graphe de visibilité, le nombre de combinaisons à tester devient $(0.0258 \times 2163)^2 \approx 3115$, soit moins de 0.07 % du nombre précédent.

Cet exemple illustre bien le principe de notre approche : d'une part la nature exacte du graphe de visibilité garantit une recherche sans perte de trajets, d'autre part il permet une forte réduction du nombre de combinaisons à tester. C'est l'impact de cette réduction sur les temps de calculs des simulations que nous évaluons dans la seconde série de tests.

Seconde série de tests : calcul de zones de couverture

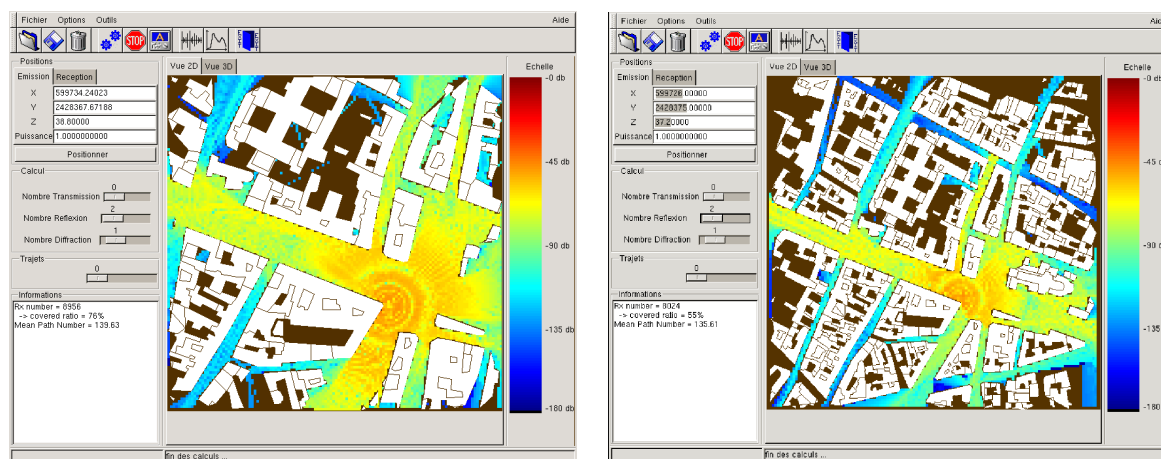


FIG. 4.8 – Calcul de zones de couverture sur la scène A (à gauche) et la scène B (à droite). Les interactions autorisées dans les deux cas sont fixées à 2 réflexions et 1 diffraction. Les variations de couleurs traduisent l'atténuation de la puissance reçue en dB.

La seconde série de tests consiste en un ensemble de calculs de zones de couverture sur les scènes A et B. Pour mieux comprendre ces résultats, il convient de donner quelques précisions sur le fonctionnement du simulateur CRT. Un calcul de couverture nécessite de trouver tous les trajets entre une position

d'émetteur donnée et toutes les positions de récepteurs fixées. La visibilité des récepteurs et de l'émetteur avec les polygones ou les dièdres d'une scène est une information précalculée (la méthode utilisée construit des arbres d'occultation pour des points de vues [BHS98]). Tout comme le graphe de visibilité, cette information est stockée dans un fichier. Pour un calcul de couverture, CRT procède comme suit :

- Il charge le graphe de visibilité et les visibilitées des récepteurs et de l'émetteur.
- Il connecte les récepteurs et l'émetteur aux nœuds du graphe en fonction des relations de visibilité précalculée.
- Pour chaque couple (émetteur, récepteur), il lance une recherche combinatoire des trajets, restreinte aux seules relations de visibilité données par le graphe.

La figure 4.8 illustre les résultats produits par CRT pour un calcul de couvertures sur les scènes A et B. Sur chaque scène, la position de l'émetteur est la même pour tous les calculs de couvertures. Sur la scène A, le nombre de récepteurs non inclus dans un bâtiment est de 8 983. Situés à une altitude de 1.5 m au dessus du sol, cela représente environ un récepteur pour 5.5 m^2 (dans le plan $z = 1.5$). Pour la scène B 8 024 récepteurs sont utilisés, soit environ un récepteur pour 15.5 m^2 . Les nombres de récepteurs sont volontairement choisis voisins pour avoir un nombre équivalent de couples (émetteur, récepteur) sur chaque scène. Le tableau 4.5 résume les temps de calcul obtenus et donne, par estimation, le temps qui serait nécessaire pour aboutir au même résultat sans utiliser le graphe de visibilité.

| Interactions | Paris A (8983 récepteurs) | | Paris B (8024 récepteurs) | |
|--------------|---------------------------|--------------|---------------------------|---------|
| | Avec GV | Sans GV | Avec GV | Sans GV |
| 1 R 0 D | 3s | 10s | 4s | 26s |
| 2 R 0 D | 36s | 2h 55min 24s | 34s | > 1j 8h |
| 2 R 1 D | 3h 18min | > 3 ans | 7h 16min | - |
| 3 R 0 D | 23min 18s | > 8 mois | 34min 23s | - |

TAB. 4.5 – Temps de calculs de zones de couvertures pour les scènes de tests A et B, en fonction du nombre et du type d'interactions fixés. Les colonnes "Avec GV" donnent les temps calculés en utilisant le graphe de visibilité, Les colonnes "Sans GV" sont les temps estimés nécessaires pour aboutir aux mêmes résultats sans un graphe de visibilité.

Ces résultats montrent clairement que l'utilisation du graphe de visibilité rend praticable des simulations autrement impossibles à obtenir sans perte de trajets. Pour chaque scène, la simulation en 2R1D, la plus coûteuse, illustre la domination du phénomène de diffraction sur celui de la réflexion. Quand une réflexion engendre un rayon, une diffraction les démultiplie, ce qui implique davantage de combinaisons à tester. D'autre part un dièdre possède un nombre supérieur de relations de visibilité qu'un polygone (voir le tableau 4.4). De ce fait, la réduction du nombre de combinaisons à tester grâce au graphe est moins importante dans le cas d'une diffraction que dans celui d'une réflexion.

4.3 Conclusions

Ce chapitre met en évidence la problématique des modèles basés rayons pour simuler la propagation des ondes électromagnétiques. À juste titre, ils sont reconnus comme les plus performants pour aider à la caractérisation du canal de propagation. Mais aucun ne peut atteindre des temps de calculs praticables sur des scènes de taille significative, sans mettre en œuvre des approches qui introduisent des pertes de trajets. De fait, le potentiel de ces modèles, en terme de précision, demeure sous-exploité. Si des simulations approchées peuvent être suffisantes pour certaines applications, la moindre perte de trajet est lourde de conséquence lorsqu'il s'agit de caractériser un canal de propagation. De plus, l'absence de simulateur "exact" pose le problème de l'évaluation des simulateurs "approximatifs" vis-à-vis du modèle.

Pour répondre à ces problèmes, nous avons présenté une optimisation sans perte de trajets qui, couplée à un simulateur basé sur la méthode des source-images, permet le traitement de scènes significatives en des temps praticables. Cette optimisation consiste en un précalcul des relations de visibilité entre les polygones et les dièdres d'un environnement donné, relations encodées dans un graphe dit "de visibilité". L'information du graphe de visibilité permet au simulateur de restreindre la recherche combinatoire des trajets aux seules primitives géométriques visibles. La nature exacte du précalcul des visibilité garantit de ne perdre aucun trajet.

Pour calculer un graphe de visibilité, il est nécessaire de traiter la visibilité de polygones mais aussi de dièdres. Nous avons montré, à travers l'analyse du bord d'un polytope représentatif de la visibilité de deux polygones, comment obtenir la visibilité de leurs arêtes. L'utilisation de l'information de visibilité représentée par un complexe de polytopes nous a permis d'unifier les calculs de visibilité entre polygones et/ou dièdres.

Pour limiter le coût total du calcul d'un graphe de visibilité, nous avons présenté une approche hiérarchique utilisant des calculs de visibilité différents adaptés au niveau de précision requis afin de contenir la complexité polynomiale du calcul du graphe.

Les résultats obtenus indiquent que l'approche présentée satisfait les objectifs. Les deux scènes de tests utilisées sont de taille significative en propagation électromagnétique. Les précalculs de leurs graphes de visibilité ne sont pas prohibitifs et permettent d'envisager de traiter des scènes plus importantes. Les calculs de zones de couverture effectués sur ces scènes montrent que le graphe de visibilité permet d'obtenir les résultats en des temps praticables quand une approche sans graphe peut nécessiter des années. A notre connaissance, notre approche constitue le premier simulateur basé rayons exact vis-à-vis du modèle de propagation utilisé. A ce titre, c'est un outil important pour la caractérisation d'un canal de propagation, mais aussi un simulateur "étalon" pour évaluer la précision de simulateurs certainement plus rapides, mais approximatifs.

4.3.1 Discussion

Plusieurs aspects de notre approche peuvent être discutés. Tout d'abord, concernant les calculs de visibilité, une requête unifiée nécessite le calcul complet du complexe de polytopes associé à deux polygones afin de pouvoir extraire la visibilité de leurs arêtes. Elle permet de traiter simultanément les visibilité de deux polygones, de deux arêtes, ou encore d'un polygone avec une arête. Comme l'indiquent les statistiques, les requêtes unifiées occupent la majeure partie du temps de calcul du graphe. Aussi, puisque seule la preuve de la visibilité est mémorisée par le graphe, on peut se demander si trois requêtes de type preuve dédiées à chacune des trois configurations permettrait d'accélérer le calcul du graphe. Une telle approche implique davantage de requêtes. Pour une scène comportant n polygones et d dièdres, elle demande $\frac{n(n-1)}{2}$ requêtes polygone-polygones, (nd) requêtes polygone-dièdre et $\frac{d(d-1)}{2}$

requêtes dièdre-dièdre. Seule une étude pratique permettrait de dire si leurs temps de calcul cumulés autorisent un gain significatif sur le calcul du graphe de visibilité.

En fait, cette problématique ne nous concerne pas, et ce pour deux raisons. La première est que le temps de précalcul du graphe n'est pas de première importance, il doit seulement ne pas être prohibitif. Ce calcul n'est nécessaire qu'une fois par scène. Contrairement au chapitre précédent, une scène utilisée en propagation n'a pas à être modifiée, et donc son graphe n'a jamais besoin d'être recalculé.

La seconde raison est qu'à moyen terme, nous souhaitons enrichir l'information de visibilité contenue par le graphe en associant à ses arcs des données supplémentaires extraites des complexes de polytopes calculés. Dans cette optique, l'utilisation de requêtes de type preuve était exclue. Nous reviendrons sur ce point dans la prochaine section.

Concernant encore le précalcul du graphe de visibilité, celui-ci n'exploite pas la cohérence spatiale puisque la visibilité de chaque feuille de la hiérarchie est traitée indépendamment des feuilles voisines. En fait, c'est un aspect que nous avons testé et qui n'a pas été retenu compte tenu de son impact quasi nul sur les temps de calcul. Le principe, adapté de [DDTP00], était de calculer les visibilités pour un nœud de la hiérarchie avant de raffiner cette information jusque dans les feuilles qu'elle contenait. Cependant le procédé ne changeait rien au nombre de requêtes unifiées à exécuter. Or, celles-ci constituent l'essentiel du coût du calcul du graphe. C'est pourquoi l'impact final sur les temps de calcul n'a jamais été significatif.

Le traitement des instabilités numériques est un aspect qui n'a pas été détaillé. Bien que l'extraction de la visibilité des arêtes ne nécessite aucun calcul numérique supplémentaire, les calculs de visibilité sont sujets au problème, comme nous l'avons déjà évoqué dans ce manuscrit. Lorsqu'une requête unifiée échoue, les primitives géométriques en jeu sont systématiquement considérées comme visibles par défaut. De la sorte, l'exactitude des simulations utilisant le graphe n'est pas remise en cause. Ces cas de figures demeurent peu nombreux. Sur les deux scènes utilisées, 12 et 35 requêtes ont échoué en raison d'instabilités numériques. C'est une quantité négligeable vis-à-vis du nombre total de requêtes (moins de 0.002 % dans les deux cas).

Enfin, les simulations réalisées n'autorisent pas un nombre important d'interactions. L'usage du graphe de visibilité réduit fortement la recherche combinatoire des trajets mais n'en modifie pas la complexité. Les temps de calculs augmentent toujours en conséquence avec le nombre d'interactions. Est-ce que cela signifie que notre approche est incapable de conserver des temps de calcul praticables avec un nombre supérieur d'interactions ? En l'état c'est certain, mais son potentiel reste sous-exploité. Ces travaux sont une première étape. En particulier, le parcours du graphe de visibilité est actuellement réalisé de manière brutale, sans tenir compte du nombre ou de la nature des interactions ou encore de la position de l'émetteur. La prise en compte de ces paramètres ouvre des perspectives permettant d'accélérer les simulations.

4.3.2 Perspectives

Comme nous venons de le souligner, l'utilisation du graphe de visibilité est actuellement rudimentaire. De nombreuses possibilités sont à explorer pour aboutir à un parcours efficace du graphe.

Une première possibilité repose sur le nombre minimal d'interactions nécessaires pour atteindre un récepteur depuis un émetteur. L'idée est d'observer qu'aucun trajet ne peut être trouvé si la longueur du plus court chemin entre l'émetteur et un récepteur est supérieure aux nombres d'interactions autorisés. En tenant compte de cette observation, le nombre de trajets à tester peut être réduit. Ce principe a déjà

fait l'objet de travaux ⁵ présentés dans [MA05].

Pour réduire encore le nombre de combinaisons à tester, la position de l'émetteur ainsi que la nature des interactions sont également à prendre en compte. Lors d'une interaction avec un élément d'un nœud du graphe de visibilité, les trajets sont recherchés parmi tous les voisins de ce nœud. Une sélection de ces voisins peut être préalablement faite afin d'éliminer ceux qui assurément ne peuvent conduire à un trajet valide. Prenons le cas de la réflexion : si l'on considère une position d'émetteur E_p (ou l'une de ses source-images) en visibilité depuis un polygone A, l'ensemble des rayons émis depuis E_p et réfléchis sur A est contenu dans une pyramide tronquée, dont le sommet est le symétrique de E_p par rapport au plan support de A (l'équivalent d'un volume d'ombre). Par conséquent, il est inutile de rechercher l'interaction suivante parmi les voisins de A qui ne sont pas contenus dans cette pyramide tronquée. Ce principe offre un moyen de réduire le nombre de voisins candidats. Il s'étend naturellement à la transmission. Dans le cas de la diffraction, la Théorie Géométrique de la diffraction établit que les rayons diffractés sont portés par le cône de Keller (voir 4.1.3). Celui-ci peut permettre de décrire la région de l'espace qui contiendra les rayons diffractés par un dièdre en visibilité de E_p . Comme pour la réflexion ou la transmission, il est inutile de rechercher l'interaction suivante parmi les voisins en dehors de cette région.

L'application de ces procédés permettrait de spécialiser le graphe de visibilité pour la position courante d'émetteur.

Enfin, comme nous l'avons précédemment évoqué, une dernière perspective est d'enrichir l'information contenue dans le graphe de visibilité, notamment en associant à chaque arc l'arbre VBSP résultant d'un calcul de visibilité. Actuellement, le graphe n'encode que la preuve de la visibilité. L'apport des arbres VBSP permettrait sa représentation complète, et offrirait de nouvelles possibilités pour rejeter toujours plus de combinaisons à tester. Par exemple, considérons un polygone B qui, sans tenir compte de l'occultation, est atteignable par réflexion sur un polygone A depuis un émetteur (ou une source-image) E_p . Maintenant, si l'on tient compte de l'occultation, il est possible qu'il n'existe pas de trajet réfléchi atteignant B en passant par A. Si tel est le cas, il est inutile de tester les combinaisons débutant par ces deux polygones. Les arbres VBSP permettraient d'éliminer ces cas de figure. Considérons les fragments f_A de A visibles depuis E_p , et E_1 sa source-image par rapport à A. Il s'agit alors de montrer que B n'est pas visible à travers les éléments de f_A depuis la source-image de E_1 . Ceci se détermine en filtrant depuis E_1 les éléments de f_A à travers l'arbre VBSP associé à A et B. Le procédé utiliserait alors la même requête point-polygone que celle présentée dans le chapitre 3. Si aucun élément de f_A demeure, B n'est pas atteignable par réflexion sur A depuis E_p . Dans le cas contraire, les éléments restants constituent exactement l'ensemble des points de A par lesquels une réflexion vers B depuis E_p est possible.

A travers cet exemple, on perçoit l'opportunité d'éliminer des trajets qu'il est inutile d'explorer, et ce en tenant compte de l'occultation, contrairement aux précédentes perspectives évoquées.

L'ensemble des perspectives présentées ici doit permettre à terme d'accélérer significativement la recherche de trajets via le graphe de visibilité. D'une approche sans perte et praticable, elles doivent permettre d'aboutir à une approche sans perte et efficace.

⁵Dans cette étude, les calculs de visibilité utilisés sont agressifs. Cependant, l'optimisation présentée est indépendante de la précision des calculs de visibilité.

Conclusion

La visibilité mutuelle de deux polygones est une question complexe et récurrente en synthèse d'images. Considérant le premier polygone comme une source surfacique, elle permet de déterminer les zones d'ombre et de pénombre projetées sur le second. Plus généralement, elle est essentielle dans les calculs d'illumination globale, où la quantité d'énergie lumineuse échangée par deux polygones dépend de leur visibilité mutuelle.

La lumière n'est pas le seul phénomène physique supportant une modélisation basée sur la notion de rayon. Aussi, comme nous l'avons vu, ces questions de visibilité dépassent le seul cadre de la synthèse d'images et trouvent leur place dès qu'il s'agit de simuler la propagation d'un flux énergétique dans une scène : en milieu homogène et isotrope, la quantité d'énergie échangée entre deux surfaces est fonction de leur visibilité. Ainsi la visibilité mutuelle de deux polygones est également importante pour les simulations acoustiques, ou encore pour la simulation des ondes électromagnétiques en radiocommunications.

Bien qu'il s'agisse d'une information importante, la plupart des méthodes se contentent de solutions approchées qui réduisent le problème à la visibilité de couples de points éléments de deux polygones. Elles doivent ensuite faire face à des problèmes d'aliasage, voir de coût en terme de temps de calcul, lorsqu'un trop grand nombre d'échantillons est nécessaire pour obtenir un résultat convaincant. Néanmoins, ce choix prédomine toujours aujourd'hui, car il est simple à mettre en œuvre. Cette popularité s'explique aussi par le fait que, pendant de longues années, ce fut la seule solution pratique connue au problème de la visibilité de deux polygones.

Cependant, au début des années 2000, des travaux basés sur la paramétrisation de Plücker ont apporté la première solution exacte permettant de prouver la visibilité mutuelle de deux polygones. Ils ont offert dans le même temps la possibilité d'en calculer une représentation.

L'idée de cette thèse était donc de repartir de ces résultats pour proposer une solution adaptée au calcul et à la représentation de la visibilité mutuelle de deux polygones, afin de démontrer tout le potentiel et l'intérêt de cette information dans la résolution de problèmes pratiques tels que le calcul des ombres douces en synthèse d'images, ou encore la propagation des ondes électromagnétiques en radiocommunications. En ce sens, nos travaux partagent l'esprit d'une approche globale ; il s'agit de calculer une structure encodant l'information de visibilité qui sera ensuite interrogée afin de répondre efficacement à des requêtes

de visibilité. Cependant, si une approche globale considère toutes les visibilitées d'un environnement, nos travaux se concentrent uniquement sur celles d'une paire de polygones.

Travaux réalisés

Pour atteindre nos objectifs, nous nous sommes tout d'abord posé le problème du calcul et de la représentation dans l'espace de Plücker de la visibilité de deux polygones. Par l'étude des précédentes approches, nous avons mis en évidence leurs difficultés quant à garantir la localité et la pertinence des opérations géométriques réalisées. Nous avons montré comment il pouvait en résulter une fragmentation de l'information de visibilité, entraînant un surcoût en terme de temps de calcul et de consommation mémoire. Un test d'intersection renforcé a été proposé pour répondre à ce problème, et intégré à notre algorithme de calcul. Celui-ci produit un arbre BSP dans l'espace de Plücker, ou arbre VBSP, qui constitue une représentation hiérarchique de la visibilité de deux polygones. L'évaluation pratique de notre algorithme a montré un gain significatif en terme de temps de calcul et d'occupation mémoire : malgré des bornes théoriques prohibitives, notre approche s'est ainsi avérée nettement moins exigeante en pratique. Disposant ainsi d'une solution adaptée au calcul et à la représentation de la visibilité de deux polygones, nous avons ensuite exploité le potentiel de cette information et proposé de nouvelles solutions dans deux domaines d'application.

En synthèse d'image, le calcul des ombres douces est toujours un sujet de recherche actif tant il apparaît difficile de concilier la rapidité et la qualité du rendu. Nous avons montré que l'information de visibilité d'une source surfacique permettait de concilier ces deux exigences. Nous avons proposé une nouvelle approche basée sur le précalcul de la visibilité de chaque polygone avec chaque source de la scène, en utilisant des arbres VBSP. Cette information permet d'extraire efficacement les parties visibles d'une source depuis un point situé sur un polygone. L'algorithme d'extraction présenté est simple : il se résume à la traversée d'un arbre VBSP, et à des calculs d'intersection d'un polygone et d'une droite en 2D. Cette solution permet le calcul analytique exact de l'illumination directe, en tous points de la scène, et pour des sources surfaciques d'émission uniforme. La qualité des ombres calculées est donc optimale. Intégrée à un logiciel de rendu, notre approche se montre clairement plus rapide qu'un logiciel stochastique commercial. Ces travaux illustrent parfaitement l'intérêt d'une structure encodant l'information de visibilité afin de répondre simplement et efficacement au problème de la visibilité des sources surfaciques depuis tous les points d'une scène.

En radiocommunications, nous avons abordé la problématique des simulateurs de propagation des ondes électromagnétiques. Nous avons expliqué pourquoi toutes les techniques actuelles procuraient seulement des résultats approchés, posant la question de la fiabilité des simulations. En réponse, nous avons présenté la première optimisation 3D sans perte de la propagation des ondes électromagnétiques. À partir de l'information de visibilité entre chaque paire de polygones de la scène, nous avons montré comment la visibilité de leurs arêtes et de leurs sommets pouvait être extraite. En utilisant cette information, nous avons présenté une architecture logicielle afin de calculer toutes les relations de visibilité entre les polygones et les dièdres d'un environnement. Encodée dans un graphe de visibilité, cette information a permis de restreindre significativement la recherche combinatoire des trajets entre un émetteur et un récepteur donné. Les tests effectués ont montré que l'on pouvait ainsi réaliser des simulations sans perte de trajets sur des scènes significatives tout en conservant des temps de calcul raisonnables.

Perspectives

Les travaux réalisés dans cette thèse nous ont permis de montrer et d'exploiter le potentiel d'une représentation dans l'espace de Plücker de la visibilité mutuelle de deux polygones. Cependant, ce potentiel est encore loin d'être épuisé et de nombreuses pistes de travail demeurent, tant au niveau du calcul de cette représentation que sur la maîtrise et l'utilisation de l'information qu'elle contient. Nous avons déjà évoqué, à la fin de chaque chapitre de ce mémoire, les perspectives soulevées par nos travaux. Nous en résumons ici les principales lignes directrices.

Nous sommes persuadés que le calcul de la visibilité doit reposer sur la silhouette étendue des objets si l'on souhaite pouvoir traiter de fins maillages de polygones. C'est une condition nécessaire pour traiter des environnements avec un niveau de détail élevé tout en maîtrisant la complexité des calculs. Plusieurs approches ont déjà montré tout l'intérêt du procédé [HMN05].

L'accroissement du niveau de détails devra également s'accompagner d'un renforcement de la robustesse des opérations géométriques réalisées dans l'espace de Plücker. Le calcul de la visibilité dans cet espace utilise plusieurs algorithmes non triviaux. Mais, comme nous l'avons souligné, ils sont finalement économes en terme d'opérations numériques. Nous avons expliqué comment le cumul des erreurs numériques pouvaient faire diverger l'algorithme de Bajaj et Pascucci [BP96]. Une solution à ce problème est de garantir une erreur maximale tout au long des calculs à l'aide d'une librairie arithmétique appropriée. A cet effet, nous souhaitons utiliser la librairie CORE⁶ [Yap04b, Yap04a] pour la prochaine évolution de notre implantation.

Associée à deux polygones d'un environnement, la validité de l'information de visibilité peut être affectée en cas de modification de la scène. En outre, un trop grand nombre de polygones peut rendre prohibitif le stockage de la visibilité de chaque paire de polygones. Pour dépasser ces limitations, une approche basée sur la visibilité relative d'un objet (*i.e.*, la visibilité des faces de sa boîte englobante), ou la visibilité d'une région nous semble une solution appropriée. La visibilité relative d'un objet doit permettre de gagner en souplesse, en supportant les isométries appliquées à n'importe quel objet dans la scène. Cette perspective offre de plus l'opportunité d'appliquer notre méthode pour le calcul d'ombres douces à des objets non déformables en mouvement. La visibilité relative d'une région doit permettre d'associer une information de visibilité à un ensemble de polygones contenus. Il ne sera plus utile de traiter individuellement chaque paire de polygones.

Les résultats obtenus pour le calcul de l'illumination directe motivent la généralisation du procédé à des calculs d'illumination globale. Pour ce faire, il est important de connaître à quel point deux polygones sont visibles. Ce qui manque actuellement, c'est une mesure fiable de la visibilité des deux polygones, ou au pire une heuristique pertinente. De plus, une telle mesure ne doit pas dépendre d'un quelconque point de vue. Par exemple, cette information sera utile pour simplifier les calculs de facteurs de formes, ou encore, pour guider le flux énergétique dans une approche de Monte Carlo couplée à une méthode d'échantillonnage d'importance. Dans les deux cas, elle doit permettre d'accélérer la convergence du résultat.

Enfin, en radiocommunications, les travaux futurs à réaliser en priorité concernent moins le calcul des visibilité que le parcours du graphe qui les encode. L'objectif que nous nous étions fixé était de garantir des simulations en des temps praticables. Compte tenu des résultats obtenus avec un parcours *brute-force* du graphe de visibilité, nous pensons pouvoir porter notre approche à un niveau de performance bien plus élevé tout en conservant la nature exacte des simulations effectuées. L'essentiel du travail consistera à "spécialiser" le graphe de visibilité pour la position de l'émetteur fixé.

⁶<http://www.cs.nyu.edu/exact/>

Pour conclure

Nous espérons avoir montré que la visibilité mutuelle de polygones n'est pas, ou tout du moins n'est plus une information trop complexe à calculer pour envisager son usage en pratique. Certes le procédé ne peut pas être considéré comme trivial. De plus, la paramétrisation de Plücker est souvent jugée peu intuitive et, en dehors des cas les plus élémentaires, son utilisation demeure relativement confidentielle. Aussi, nous espérons que ce mémoire contribuera à populariser ce type d'approche.

Enfin, nous pensons avoir montré comment cette information pouvait être exploitée pour imaginer des réponses nouvelles, simples et efficaces, à des questions pratiques. Mais, au delà des travaux présentés dans cette thèse, nous avons aussi souligné tout le potentiel restant à exploiter, en espérant que celui-ci suscitera l'intérêt de nombreuses personnes à l'avenir.

Bibliographie

- [AAM03] Ulf Assarsson and Tomas Akenine-Möller. A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Trans. Graph.*, 22(3) :511–520, 2003. 68
- [AAM04] Timo Aila and Tomas Akenine-Möller. A hierarchical shadow volume algorithm. In *Proceedings of Graphics Hardware 2004*, pages 15–23. Eurographics Association, 2004. 67
- [AC04] L. Aveneau and P. Combeau. Path finding based on monte carlo techniques compared with a full ray-tracing approach in narrow and wide bands. Mai 2004. VTC2004, Milan, Italia. 91
- [ADMAM03] Ulf Assarsson, Michael Dougherty, Michael Mounier, and Tomas Akenine-Möller. An optimized soft shadow volume algorithm with real-time performance. In *HWWS '03 : Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 33–40, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 68
- [AF96] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Appl. Math.*, 65(1-3) :21–46, 1996. 34
- [AFF97] F. A. Agelet, F. P. Fontan, and A. Formella. Fast Ray-Tracing for Microcellular and Indoor Environments. *IEEE Trans. Magn.*, 33 :1484–1487, March 1997. 91
- [AFR⁺00] Agelet, Formella, Rabanos, Vicente, and Fontan. Efficient Ray-Tracing Acceleration Techniques for Radio Propagation Modeling. *IEEE Transactions on Vehicular Technology*, 49 :2089–2104, November 2000. 91
- [AL04] Timo Aila and Samuli Laine. Alias-free shadow maps. In *Proceedings of Eurographics Symposium on Rendering 2004*, pages 161–166. Eurographics Association, 2004. 67
- [Ama84] John Amanatides. Ray tracing with cones. In *SIGGRAPH '84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 129–135, New York, NY, USA, 1984. ACM Press. 64, 89
- [AMA02] Tomas Akenine-Möller and Ulf Assarsson. Approximate soft shadows on arbitrary surfaces using penumbra wedges. In *EGRW '02 : Proceedings of the 13th Eurographics workshop on Rendering*, pages 297–306, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association. 68

- [APS93] Boris Aronov, Marco Pellegrini, and Micha Sharir. On the zone of a surface in a hyperplane arrangement. *Discrete Comput. Geom.*, 9(2) :177–186, 1993. 26
- [APVM00] L. Aveneau, Y. Pousset, R. Vauzelle, and M. Mériaux. Development and evaluations of physical and computer optimizations for the utd 3d model. Avril 2000. IEEE Antennas & Propagation'2000, Davos, Suisse. 90
- [ARFPB90] John M. Airey, John H. Rohlfs, and Jr. Frederick P. Brooks. Towards image realism with interactive update rates in complex virtual building environments. In *SI3D '90 : Proceedings of the 1990 symposium on Interactive 3D graphics*, pages 41–50, New York, NY, USA, 1990. ACM Press. 33
- [Arv94] James Arvo. The irradiance jacobian for partially occluded polyhedral sources. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 343–350, New York, NY, USA, 1994. ACM Press. 67
- [Ave99] L. Aveneau. *Phénomènes Ondulatoires en Synthèse d'Images*. PhD thesis, Décembre 1999. Thèse de Doctorat de l'Université de Poitiers. 90
- [AWG78] Peter Atherton, Kevin Weiler, and Donald Greenberg. Polygon shadow generation. In *SIGGRAPH '78 : Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 275–281, New York, NY, USA, 1978. ACM Press. 62
- [BDD⁺02] Herve Brönnimann, Olivier Devillers, Vida Dujmovic, Hazel Everett, Marc Glisse, Xavier Goaoc, Sylvain Lazard, Hyeon-Suk Na, and Sue Whitesides. On the number of lines tangent to four convex polyhedra. In *14th Canadian Conference on Computational Geometry - CCCG'02, Lethbridge, Canada, Aug 2002*. 29
- [BDD⁺04] Herve Brönnimann, Olivier Devillers, Vida Dujmovic, Hazel Everett, Marc Glisse, Xavier Goaoc, Sylvain Lazard, Hyeon-Suk Na, and Sue Whitesides. The number of lines tangent to arbitrary convex polyhedra in 3d. In *ACM Symposium on Computational Geometry - SoCG'2004, Brooklyn, NY, Jun 2004*. 29, 56
- [Ber86] W. Bergeron. A general version of crow's shadow volumes. *IEEE Computer Graphics and applications*, 6(9) :17–28, 1986. 62
- [BHS98] J. Bittner, V. Havran, and P. Slavík. Hierarchical visibility culling with occlusion trees. In *Proceedings of Computer Graphics International '98 (CGI'98)*, pages 207–219. IEEE, 1998. 33, 34, 62, 107
- [Bit02] J. Bittner. *Hierarchical Techniques for Visibility Computations*. PhD thesis, Czech Technical University in Prague, October 2002. 2, 12, 23, 33, 39, 40, 58, 82, 100
- [BM94] B. Bouche and F. Molinet. *Méthodes asymptotiques en électromagnétisme*, volume 1^{re} édition. Springer Verlag, Berlin, 1994. 88
- [BP96] C. L. Bajaj and V. Pascucci. Splitting a complex of convex polytopes in any dimension. In *In Proceedings of the 12th Annual Symposium on Computational Geometry*, pages 88–97. ACM, May 1996. 32, 35, 53, 57, 58, 82, 113
- [BP01] Jiří Bittner and Jan Příkryl. Exact regional visibility using line space partitioning. Technical Report TR-186-2-01-06, Karlsplatz 13/186/2, A-1040 Vienna, Austria, 2001. 33
- [BW97] M. Born and E. Wolf. *Principles of Optics*, volume 6^{me} édition. Pergamon Press, New York, 1997. 88

- [BW03] Jiří Bittner and Peter Wonka. Visibility in computer graphics. Technical Report TR-186-2-03-03, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, march 2003. human contact : technical-report@cg.tuwien.ac.at. 11
- [BWW05] Jiří Bittner, Peter Wonka, and Michael Wimmer. Fast exact from-region visibility in urban scenes. In Kavita Bala and Philip Dutré, editors, *Rendering Techniques 2005 (Proceedings of the Eurographics Symposium on Rendering 2005)*, pages 223–230. Eurographics, Eurographics Association, June 2005. 33, 100
- [CF89] Norman Chin and Steven Feiner. Near real-time shadow generation using bsp trees. In *SIGGRAPH '89 : Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 99–106, New York, NY, USA, 1989. ACM Press. 62, 66
- [CF92] Norman Chin and Steven Feiner. Fast object-precision shadow generation for area light sources using bsp trees. In *SI3D '92 : Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 21–30, New York, NY, USA, 1992. ACM Press. 66, 67
- [Cha97] G. Chartier. *Manuel d'Optique*. Hermès, Paris, 1997. 88
- [CJP98] M. F. Catedra and O. Gutierrez J. Perez, F. S. de Anana. Efficient Ray-Tracing Techniques for Three-Dimensional Analyses of Propagation in Mobile Communications : Application to Picocell and Microcell Scenarios. *IEEE Antennas Propagat. Mag.*, 40 :15–28, April 1998. 91
- [COCD03] Cohen-Or, Chrysanthou, and Silva Durand. A survey of visibility for walkthrough applications. In *IEEE TVCG*, pages 412–431, july-september 2003. 100
- [Coo86] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5(1) :51–72, 1986. 63
- [CPC84] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *SIGGRAPH '84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 137–145, New York, NY, USA, 1984. ACM Press. 63
- [Cro77] Franklin C. Crow. Shadow algorithms for computer graphics. In *SIGGRAPH '77 : Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 242–248, New York, NY, USA, 1977. ACM Press. 62, 65
- [CS95] Yiorgos Chrysanthou and Mel Slater. Shadow volume bsp trees for computation of shadows in dynamic scenes. In *SI3D '95 : Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 45–50, New York, NY, USA, 1995. ACM Press. 62
- [CT96] S. Coorg and S. Teller. A spatially and temporally coherent object space visibility algorithm. Technical report, 1996. 46
- [CT97] Satyan Coorg and Seth Teller. Real-time occlusion culling for models with large occluders. In *SI3D '97 : Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 83–ff., New York, NY, USA, 1997. ACM Press. 46
- [dABD⁺00] F. S. de Adana, O. G. Blonco, I. G. Diego, J. P. Arriaga, and M. F. Catedra. Propagation Model Based on Ray Tracing for the Design of Personal Communication Systems in Indoor Environments. *IEEE-VEH*, 49 :2105–2112, November 2000. 91
- [DD02] Florent Duguet and George Drettakis. Robust epsilon visibility. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 567–575, New York, NY, USA, 2002. ACM Press. 30, 69

- [DDP97] Frédo Durand, George Drettakis, and Claude Puech. The visibility skeleton : a powerful and efficient multi-purpose global visibility tool. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 89–100. ACM Press/Addison-Wesley Publishing Co., 1997. 29, 69
- [DDP98] Frédo Durand, George Drettakis, and Claude Puech. Visibility driven hierarchical radiosity. In *SIGGRAPH '98 : ACM SIGGRAPH 98 Conference abstracts and applications*, page 263, New York, NY, USA, 1998. ACM Press. 29, 69, 83
- [DDP99] Frédo Durand, George Drettakis, and Claude Puech. Fast and accurate hierarchical radiosity using global visibility. *ACM Trans. Graph.*, 18(2) :128–170, 1999. 29, 69, 83
- [DDP02] Frédo Durand, George Drettakis, and Claude Puech. The 3d visibility complex. *ACM Trans. Graph.*, 21(2) :176–206, 2002. 29
- [DDTP00] Frédo Durand, George Drettakis, Joelle Thollot, and Claude Puech. Conservative visibility preprocessing using extended projections. *Proceedings of SIGGRAPH 2000*, July 2000. Held in New Orleans, Louisiana. 109
- [DF94] George Drettakis and Eugene Fiume. A fast shadow algorithm for area light sources using backprojection. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 223–230, New York, NY, USA, 1994. ACM Press. 69
- [DORP96] Frédo Durand, Rachel Orti, Stéphane Rivière, and Claude Puech. Radiosity in flatland made visibly simple : using the visibility complex for lighting simulation of dynamic scenes in flatland. In *SCG '96 : Proceedings of the twelfth annual symposium on Computational geometry*, pages 511–512, New York, NY, USA, 1996. ACM Press. 29
- [Dug04] Florent Duguet. Shadow computations using robust epsilon visibility. Technical Report RR-5167, INRIA, REVES/INRIA Sophia-Antipolis, April 2004. 30
- [Dur99] F. Durand. *3D Visibility, analysis and applications*. PhD thesis, University Joseph Fourier, 1999. 11, 27, 29
- [DWBS03] Andreas Dietrich, Ingo Wald, Carsten Benthin, and Philipp Slusallek. The OpenRT Application Programming Interface – Towards A Common API for Interactive Ray Tracing. In *Proceedings of the 2003 OpenSG Symposium*, pages 23–31, Darmstadt, Germany, 2003. Eurographics Association. 64
- [EBD92] Eggert, Bowyer, and Dyer. Aspect graphs : State-of-the-art and applications in digital photogrammetry. In *Proceedings of the 17th Congress of the International Society for Photogrammetry and Remote Sensing, Part B5*, pages 633–645, 1992. 27
- [EK03] Cass Everitt and Mark J Kilgard. Practical and robust stenciled shadow volumes for hardware-accelerated rendering, 2003. 62
- [EOS86] H Edelsbrunner, J O'Rourke, and R Seiel. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.*, 15(2) :341–363, 1986. 26
- [FR94] Komei Fukuda and Vera Rosta. Combinatorial face enumeration in convex polytopes. *Comput. Geom. Theory Appl.*, 4(4) :191–198, 1994. 45, 52
- [GCS91] Ziv Gigus, John Canny, and Raimund Seidel. Efficiently computing and representing aspect graphs of polyhedral objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(6) :542–551, 1991. 27
- [GH98] Djamchid Ghazanfarpour and Jean-Marc Hasenfratz. A beam tracing with precise antialiasing for polyhedral scenes. *Computer Graphics*, 22(1) :103–115, 1998. 64, 89

- [GM90] Z. Gigus and J. Malik. Computing the aspect graph for line drawings of polyhedral objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(2) :113–122, 1990. 27
- [Goa04] Xavier Goaoc. *Structures de visibilité globales : tailles, calculs et dégénérescences*. Thèse d’université, Université Nancy 2, Mai 2004. 29, 56
- [Hai00] Eric Haines. A shaft culling tool. *J. Graph. Tools*, 5(1) :23–26, 2000. 53
- [Hav00] Vlastimil Havran. *Heuristic Ray Shooting Algorithms*. Ph.d. thesis, Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, November 2000. 64
- [Hec92] Paul S. Heckbert. Discontinuity meshing for radiosity. In *Third Eurographics Workshop on Rendering*, pages 203–216, Bristol, UK, May 1992. 69
- [Hei91] Tim Heidmann. Real shadows, real time. *Iris Universe*, 18 :28–31, 1991. Silicon Graphics, Inc. 62
- [HG86] Eric Haines and Donald P. Greenberg. The light buffer :a ray tracer shadow testing accelerator. *IEEE Computer Graphics and Applications*, 9(6) :6–16, 1986. 64, 91
- [HH84] Paul S. Heckbert and Pat Hanrahan. Beam tracing polygonal objects. In *SIGGRAPH ’84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 119–127, New York, NY, USA, 1984. ACM Press. 64
- [HJGZ97] Martin Henk, Jürgen Richter-Gebert, and Günter Ziegler. *Handbook of Discrete and Computational Geometry*, pages 243–270. CRC Press, 1997. Basic properties of convex polytopes. 17
- [HK85] Martial Hebert and Takeo Kanade. The 3-d profile method for object recognition. In *Proceedings of the 1985 Computer Vision and Pattern Recognition Conference (CVPR ’85)*, pages 458–464, 1985. 27
- [HLHS03] Jean-Marc Hasenfratz, Marc Lapierre, Nicolas Holzschuch, and François Sillion. A survey of real-time soft shadows algorithms. *Computer Graphics Forum*, 22(4) :753–774, Dec. 2003. State-of-the-Art Reviews. 62
- [HMN05] Denis Haumont, Otso Mäkinen, and Shaun Nirenstein. A low dimensional framework for exact polygon-to-polygon occlusion queries. In *Proceedings of the Eurographics Symposium on Rendering*, pages 211–222, June 2005. 32, 35, 45, 52, 58, 81, 92, 99, 100, 101, 113
- [HP03] Vlastimil Havran and Werner Purgathofer. On comparing ray shooting algorithms. *Computers & Graphics*, 27(4) :593–604, 2003. 64
- [HPP00] Vlastimil Havran, Jan Prikryl, and Werner Purgathofer. Statistical comparison of ray-shooting efficiency schemes. Technical Report TR-186-2-00-14, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritens-trasse 9-11/186, A-1040 Vienna, Austria, May 2000. human contact : technical-report@cg.tuwien.ac.at. 64, 100
- [HWL99] Hoppe, Wölfle, and Landstorfer. Fast 3d ray tracing for the planning of microcells by intelligent preprocessing of the database. March 1999. 91
- [IY02] M. F. Iskander and Z. Y. Yun. Propagation Prediction Models for Wireless Communication Systems. *IEEE Transactions on Microwave Theory and Techniques*, 50(3), Mars 2002. 88
- [Kel62] J. B. Keller. Geometrical Theory of Diffraction. *J. Opt. Soc. Amer.*, 52 :116–130, 1962. 88

- [KP74] R. G. Kouyoumjian and P. H. Pathak. A Uniform Geometrical Theory of Diffraction for an Edge in a Perfectly Conducting Surface. volume 62, pages 1448–1461, November 1974. 88
- [KP02] Volker Kaibel and Marc E. Pfetsch. Computing the face lattice of a polytope from its vertex-facet incidences. *Computational Geometry : Theory and Applications*, 23(3) :281–290, November 2002. 52
- [KvD76] J. Koenderink and A.J. van Doorm. The singularities of visual mapping. *Biological Cybernetics*, 24(1) :51–59, 1976. 26
- [KvD79] J. Koenderink and A.J. van Doorm. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32(1) :211–216, 1979. 26
- [LA05] Samuli Laine and Timo Aila. A hierarchical penumbra casting. In *Proceedings of Eurographics 2005*, pages 00–00. Eurographics Association, 2005. 67
- [LAA⁺05] Samuli Laine, Timo Aila, Ulf Assarsson, Jaakko Lehtinen, and Tomas Akenine-Möller. Soft shadow volumes for ray tracing. In *Proceedings of ACM Siggraph 2005*, pages 00–00. ACM, 2005. 68
- [LAM01] Jonas Lext and Tomas Akenine-Möller. Towards rapid reconstruction for animated ray tracing. In *Proceedings of Eurographics 2001, Short Presentations*, 2001. 83
- [LCL89] H. Ling, R. Chou, and S. Lee. Shooting and Bouncing Rays : Calculating the RCS of an Arbitrarily Shaped Cavity. *IEEE Transactions on Antennas and Propagation*, 37 :194–205, January 1989. 89
- [LWGM04] Brandon Lloyd, Jeremy Wend, Naga K. Govindaraju, and Dinesh Manocha. Cc shadow volumes. In *Rendering Techniques*, pages 197–206, 2004. 62
- [MA05] F. Mora and L. Aveneau. Optimised scanning of a visibility graph data structure for efficient ray-tracing. Octobre 2005. Proceedings of ECWT2005, Paris. 91, 110
- [MAM05] F. Mora, L. Aveneau, and M. Mériaux. Coherent and exact polygon-to-polygon visibility. In *Proceedings of WSCG'05*, 2005. 39
- [McG04] Morgan McGuire. Observations on silhouette sizes. *jgt*, 9(1) :1–12, 2004. 36, 68, 81
- [MP96] David M. Mount and Fan-Tao Pu. Stabbing orthogonal objects in 3-space. Technical Report CS-TR-3701, 1996. 59
- [MP99] David M. Mount and Fan-Tao Pu. Binary space partitions in plücker space. In *ALENEX '99 : Selected papers from the International Workshop on Algorithm Engineering and Experimentation*, pages 94–113, London, UK, 1999. Springer-Verlag. 30
- [NAT90] Naylor, Amanatides, and Thibault. Merging bsp trees yields polyhedral set opérations. In *In Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 115–254. ACM Press, 1990. 33
- [Nay90] B. Naylor. Binary space partitioning trees as an alternative representation of polytopes. *Comput. Aided Des.*, 22(4) :250–252, 1990. 33
- [NBG02] S. Nirenstein, E. Blake, and J. Gain. Exact from-region visibility culling. In *Proceedings of the 13th Eurographics workshop on Rendering*, pages 191–202. Eurographics Association, June 2002. 2, 11, 23, 31, 39, 40, 103
- [Nir03] S. Nirenstein. *Fast and accurate visibility preprocessing*. PhD thesis, University of Cap Town, South Africa, October 2003. 12, 31, 52, 58
- [NN72] Tomoyuki Nishita and Eihachiro Nakamae. An algorithm for hidden line elimination of polyhedra. *Information Processing in Japan*, 12 :134–141, 1972. 62

- [NN74] Tomoyuki Nishita and Eihachiro Nakamae. An algorithm for half toned representation of three dimensional objects. *Information Processing in Japan*, 14(1) :93–99, 1974. 62, 65
- [NN83] Tomoyuki Nishita and Eihachiro Nakamae. Half-tone representation of 3-d objects illuminated by area sources or polyhedron sources. In *Proceedings of IEEE Computer Software and Application Conference*, pages 237–342, 1983. 65, 67, 74
- [NN85] Tomoyuki Nishita and Eihachiro Nakamae. Continuous tone representation of three-dimensional objects taking account of shadows and interreflection. In *SIGGRAPH '85 : Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 23–30, New York, NY, USA, 1985. ACM Press. 65, 66, 67, 74
- [NON85] Tomoyuki Nishita, Isao Okamura, and Eihachiro Nakamae. Shading models for point and linear sources. *ACM Trans. Graph.*, 4(2) :124–146, 1985. 65
- [ORDP96] Rachel Orti, Stéphane Rivière, Frédo Durand, and Claude Puech. Radiosity for dynamic scenes in flatland with the visibility complex. In Jarek Rossignac and François Sillion, editors, *Computer Graphics Forum (Proc. of Eurographics '96)*, volume 16, pages 237–249, Poitiers, France, Sep 1996. 29
- [PD86] W. H. Plantinga and C. R. Dyer. An algorithm for computing the aspect graph. In *Proceedings of the 27th Symposium on the Foundations of Computer Science*, pages 123–131. IEEE Computer Society Press, October 1986. 27
- [PD90] W. H. Plantinga and C. R. Dyer. Visibility, occlusion and the aspect graph. *International Journal of Computer vision*, 5(2) :137–160, 1990. 27, 83
- [PDS90] H. Plantinga, C. R. Dyer, and W. B. Seales. Real-time hidden-line elimination for a rotating polyhedral scene using the aspect representation. In *Proceedings on Graphics interface '90*, pages 9–16, Toronto, Ont., Canada, Canada, 1990. Canadian Information Processing Society. 27
- [Pel90] Marco Pellegrini. Stabbing and ray shooting in 3 dimensional space. In *SCG '90 : Proceedings of the sixth annual symposium on Computational geometry*, pages 177–186, New York, NY, USA, 1990. ACM Press. 26
- [Pel91] Marco Pellegrini. *Combinatorial and algorithmic analysis of stabbing and visibility problems in three-dimensional space*. PhD thesis, New York, NY, USA, 1991. 26
- [Pel93] Marco Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica*, 9(5) :471–494, 1993. 26
- [Pel97] Marco Pellegrini. *Handbook of Discrete and Computational Geometry*. CRC Press, 1997. Ray shooting and lines in space, pages 599-614. 26
- [Pla88] William Harry Plantinga. *The asp : a continuous, viewer-centered object representation for computer vision*. PhD thesis, 1988. Adviser-Charles R. Dyer. 27, 83
- [PPS04] A. Paoluzzi, V. Pascucci, and G. Scorzelli. Progressive dimension-independent boolean operations. In *Proceeding of the 9-th ACM Symposium on Solid Modeling and Applications*, pages 203–211. ACM, June 2004. 57
- [Pu98] Fan-Tao Pu. *Data structures for global illumination computation and visibility queries in 3-space*. PhD thesis, 1998. Chair-David M. Mount. 30, 59
- [PV93] Michel Pocchiola and Gert Vegter. The visibility complex. In *Proceedings of the ninth annual symposium on Computational geometry*, pages 328–337. ACM Press, 1993. 28, 29

- [PV96] M. Pocchiola and G. Vegter. Topologically sweeping visibility complexes via pseudo-triangulations. *Discrete Computational Geometry*, 16(4) :419–453, 1996. 28
- [Riv95] Stéphane Rivière. Topologically sweeping the visibility complex of polygonal scenes. In *11th Annual ACM Symposium on Computational Geometry*, 1995. 28
- [Riv97a] Stéphane Rivière. Dynamic visibility in polygonal scenes with the visibility complex. In *13th Annual ACM Symposium on Computational Geometry*, 1997. 29
- [Riv97b] Stéphane Rivière. Walking in the visibility complex with applications to visibility polygons and dynamic visibility. In *9th Canadian Conference on Computational Geometry (CCCG97)*, Kingston, Canada, Aug 1997. 29
- [SG94] A. James Stewart and Sherif Ghali. Fast computation of shadow boundaries using spatial coherence and backprojections. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 231–238, New York, NY, USA, 1994. ACM Press. 69
- [SM97] H. Suzuki and A. S. Mohan. Ray Tube Tracing Method for Predicting Indoor Channel Characteristic Map. *Electron. Lett.*, 33(17) :1495–1496, 1997. 89
- [Som59] Sommerville. *Analytical Geometry in Three Dimension*. Cambridge University Press, 1959. 16
- [SR94] S. Y. Seidel and T. S. Rappaport. Site-Specific Propagation Prediction for Wireless In-Building Personal Communication System Design. *IEEE-VEH*, 43 :879–891, November 1994. 89
- [SSSK04] László Szécsi, Mateu Sbert, and László Szirmay-Kalos. Correlated and importance sampling in direct light source computation and environment mapping. In *Proceedings of Eurographics'04*, 2004. 64
- [STN87] Mikio Shinya, T. Takahashi, and Seiichiro Naito. Principles and applications of pencil tracing. *SIGGRAPH Comput. Graph.*, 21(4) :45–54, 1987. 64
- [Sto91] Jorge Stolfi. *Oriented Projective Geometry*. Academic Press, Boston, 1991. 13, 16
- [SWS05] Jörg Schmittler Sven Woop and Philipp Slusallek. Rpu : A programmable ray processing unit for realtime ray tracing. In *Proceedings of ACM SIGGRAPH 2005*, July 2005. 64
- [SWZ96] Peter Shirley, Changyaw Wang, and Kurt Zimmerman. Monte carlo techniques for direct lighting calculations. *ACM Trans. Graph.*, 15(1) :1–36, 1996. 64
- [Tel92a] Seth Teller. *Visibility Computation in Densely Occluded Polyhedra Environnements*. PhD thesis, University of California, Berkeley, 1992. 27, 65
- [Tel92b] Seth J. Teller. Computing the antipenumbra of an area light source. *Computer Graphics*, 26(2) :139–148, 1992. 21, 27, 28, 100
- [TH91] Seth Teller and Michael E. Hohmeyer. Stabbing oriented convex polygons in randomized $O(n^2)$ time. Technical report, Technical Report UCB/CSD 91/669, CS Department, UC Berkeley, July 1991. 21, 27, 28, 100
- [TH93] Seth Teller and Pat Hanrahan. Global visibility algorithms for illumination computations. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 239–246. ACM Press, 1993. 21, 28, 47
- [TH99] Seth Teller and Michael Hohmeyer. Determining the lines through four lines. *Journal of Graphics Tools*, 4(3) :11–22, 1999. 9, 27

- [TT96] S. Y. Tan and H. S. Tan. A Microcellular Communications Propagation Model Based on the Uniform Theory of Diffraction and Multiple Image Theory. *IEEE trans. Antennas Propagation*, 44 :1317–1326, October 1996. 90
- [TT97] Toshimitsu Tanaka and Tokiichiro Takahashi. Fast analytic shading and shadowing for area light sources. In *Computer Graphics Forum (Eurographics '97)*, volume 16, pages 231–240, September 1997. 66, 82
- [WA77] Kevin Weiler and Peter Atherton. Hidden surface removal using polygon area sorting. In *SIGGRAPH '77 : Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 214–222, New York, NY, USA, 1977. ACM Press. 62
- [Wal04] Ingo Wald. *Realtime Ray Tracing and Interactive Global Illumination*. PhD thesis, Computer Graphics Group, Saarland University, 2004. Available at <http://www.mpi-sb.mpg.de/~wald/PhD/>. 64
- [WBS02] Ingo Wald, Carsten Benthin, and Philipp Slusallek. OpenRT – A Flexible and Scalable Rendering Engine for Interactive 3D Graphics. submitted for publication, meanwhile available as a Technical Report, TR-2002-01, Saarland University, 2002. 64
- [WBS03] Ingo Wald, Carsten Benthin, and Philipp Slusallek. Distributed Interactive Ray Tracing of Dynamic Scenes. In *Proceedings of the IEEE Symposium on Parallel and Large-Data Visualization and Graphics (PVG)*, 2003. 83
- [WBWS01] Ingo Wald, Carsten Benthin, Markus Wagner, and Philipp Slusallek. Interactive rendering with coherent ray tracing. In Alan Chalmers and Theresa-Marie Rhyne, editors, *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2001)*, volume 20. Blackwell Publishers, Oxford, 2001. available at <http://graphics.cs.uni-sb.de/wald/Publications>. 64
- [Whi80] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6) :343–349, 1980. 63
- [Yap04a] Chee Yap. On guaranteed accuracy computation. In Falai Chen and Dongming Wang, editors, *Geometric Computation (tentative)*. World Scientific Press, 2004. 58, 113
- [Yap04b] Chee K. Yap. Robust geometric computation. In J. E. Goodman and J. O'Rourke, editors, *CRC Handbook in Computational Geometry*. CRC Press, 2004. 58, 113
- [YIZ01] Z. Yun, M. F. Iskander, and Z. Zhang. Developpment of a New Shooting-and-Bouncing Ray (SBR) Tracing Method That Avoids Ray Double Counting. *Antennas and Propagation Society International Symposium*, 1 :464–467, 2001. 89